Cat.No. W318–E1–4

CompoBus/S

SRM1(-V2)

Master Control Units

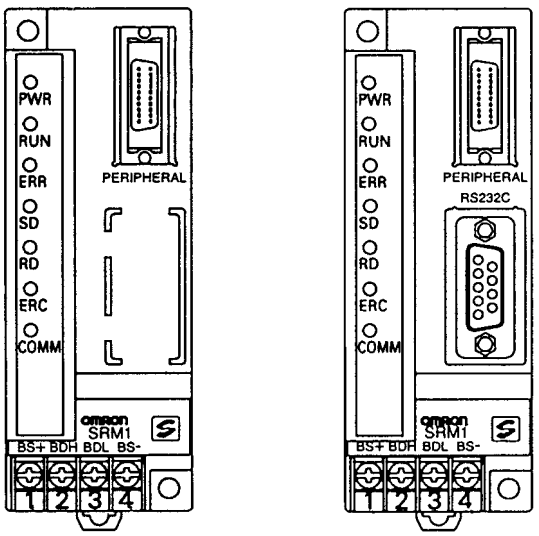
OPERATION MANUAL

OMRON

CompoBus/S
SRM1(-V2) Master Control Units

Operation Manual

Revised May 2000



Notice:

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify precautions in this manual. Always heed the information provided with them. Failure to heed precautions can result in injury to people or damage to property.

⚠ DANGER Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ Caution Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

OMRON Product References

All OMRON products are capitalized in this manual. The word “Unit” is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation “Ch,” which appears in some displays and on some OMRON products, often means “word” and is abbreviated “Wd” in documentation in this sense.

The abbreviation “PC” means Programmable Controller and is not used as an abbreviation for anything else.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3... 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

© OMRON, 1997

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

TABLE OF CONTENTS

PRECAUTIONS xi

1 Intended Audience xii
2 General Precautions xii
3 Safety Precautions xii
4 Operating Environment Precautions xiii
5 Application Precautions xiii

SECTION 1
Introduction 1

1-1 SRM1 Features and Functions 2
1-2 System Configuration 3
1-3 Procedures From System Design to Test Operation 7
1-4 I/O and Data Area Allocations 8

SECTION 2
Specifications and Components 11

2-1 Specifications 12
2-2 Unit Components 15

SECTION 3
Installation and Wiring 17

3-1 System Design 18
3-2 Selecting an Installation Site 19
3-3 Installing the SRM1 20
3-4 Wiring and Connections 21
3-5 One-to-one NT Link 30
3-6 One-to-N NT Link 30
3-7 One-to-one PC Link Connections 31

SECTION 4
Using the Programming Console 33

4-1 Basic Operations 34
4-2 Programming Console Operations 36

SECTION 5
Test Runs and Error Processing 59

5-1 Startup Procedure 60
5-2 Entering the Program 62
5-3 Test Run 71
5-4 Error Processing 72
5-5 Programming Console Operation Errors 75
5-6 Programming Errors 75
5-7 Troubleshooting Flowcharts 77

SECTION 6
Expansion Memory Unit 83

6-1 Overview 84
6-2 Specifications and Nomenclature 85
6-3 Handling 86

TABLE OF CONTENTS

Appendices

A Standard Models 93
B External Dimensions 97

Glossary 99

Index 115

Revision History 119

About this Manual:

The SRM1 is a special CompoBus/S controller that provides remote I/O with greatly reduced wiring. A distributed I/O system with up to 32 Slaves and 256 I/O points can be constructed. There are two manuals describing the setup and operation of the SRM1: The *SRM1(-V2) Operation Manual* (this manual) and the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual* (W353).

This manual describes the system configuration and installation of the SRM1 and provides a basic explanation of operating procedures for the Programming Consoles and introduces the capabilities of the SYSMAC Support Software (SSS). Read this manual first to acquaint yourself with the SRM1.

The *CompoBus/S Operation Manual* (W266) provides descriptions of the CompoBus/S system and Units.

The *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual* (W353) provides detailed descriptions of the SRM1's programming functions. The *SYSMAC Support Software (SSS) Operation Manuals: Basics* (W247) *and C-series PCs* (W248) provide descriptions of SSS operations for the SRM1 and C-series PCs.

The *SYSMAC-CPT Support Software Quick Start Guide* (W332) and *User Manual* (W333) provide descriptions of ladder diagram operations in the Windows environment.

The *WS02-CXPC1-E CX-Programmer User Manual* (W361) and the *CX-Server User Manual* (W362) provide details of operations for the WS02-CXPC1-E CX-Programmer.

Please read this manual carefully and be sure you understand the information provide before attempting to install and operate the SRM1.

Section 1 describes the SRM1's special features and functions and shows the possible system configurations.

Section 2 provides the technical specifications of the SRM1 and describes its main components.

Section 3 explains how to install and wire the SRM1. Be sure to follow the instructions contained here concerning the control panel, power supply, CompoBus/S transmissions, and RS-232C Port wiring.

Section 4 explains how to use the Programming Console. Be sure to read this section carefully if you are not already familiar with Programming Console operations.

Section 5 describes procedures for trial runs of SRM1 operation, self-diagnosis functions, and error processing to identify and correct the hardware and software errors that can occur during operation.

Section 6 describes how to use the CPM1-EMU01-V1 Expansion Memory Unit. Follow the handling precautions and procedures to properly use the Unit.

Appendix A provides a list of standard models.

Appendix B provides the external dimensions.

⚠WARNING Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each section in its entirety and be sure you understand the information provided in the section and related sections before attempting any of the procedures or operations given.

PRECAUTIONS

This section provides general precautions for using the SRM1 and related devices.

The information contained in this section is important for the safe and reliable application of the SRM1. You must read this section and understand the information contained before attempting to set up or operate a CompoBus/S System.

1 Intended Audience xii
2 General Precautions xii
3 Safety Precautions xii
4 Operating Environment Precautions xiii
5 Application Precautions xiii

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of installing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.

Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.

Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.

This manual provides information for programming and operating the OMRON SRM1. Be sure to read this manual before attempting to use the software and keep this manual close at hand for reference during operation.

⚠ WARNING It is extremely important that an SRM1 and all CompoBus/S Units be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying a CompoBus/S System to the abovementioned applications.

3 Safety Precautions

⚠ WARNING Never attempt to disassemble any Units while power is being supplied. Doing so may result in serious electrical shock or electrocution.

⚠ WARNING Never touch any of the terminals while power is being supplied. Doing so may result in serious electrical shock or electrocution.

⚠ WARNING Provide safety measures in external circuits (i.e., not in the Programmable Controller), including the following items, in order to ensure safety in the system if an abnormality occurs due to malfunction of the PC or another external factor affecting the PC operation. Not doing so may result in serious accidents.

- Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.
- The PC will turn OFF all outputs when its self-diagnosis function detects any error or when a severe failure alarm (FALS) instruction is executed. As a countermeasure for such errors, external safety measures must be provided to ensure safety in the system.
- The PC outputs may remain ON or OFF due to deposition or burning of the output relays or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.

⚠ WARNING When transferring programs to other nodes, or when making changes to I/O memory, confirm the safety of the destination node before transfer. Not doing so may result in injury.

⚠ Caution Execute online edit only after confirming that no adverse effects will be caused by extending the cycle time. Otherwise, the input signals may not be readable.

4 Operating Environment Precautions

Do not operate the control system in the following places.

- Where the SRM1 is exposed to direct sunlight.
- Where the ambient temperature is below 0°C or over 55°C.
- Where the SRM1 may be affected by condensation due to radical temperature changes.
- Where the ambient humidity is below 10% or over 90%.
- Where there is any corrosive or inflammable gas.
- Where there is excessive dust, saline air, or metal powder.
- Where the SRM1 is affected by vibration or shock.
- Where any water, oil, or chemical may splash on the SRM1.

⚠ Caution The operating environment of the CompoBus/S System can have a large effect on the longevity and reliability of the system. Improper operating environments can lead to malfunction, failure, and other unforeseeable problems with the CompoBus/S System. Be sure that the operating environment is within the specified conditions at installation and remains within the specified conditions during the life of the system.

5 Application Precautions

Observe the following precautions when using the SRM1.

⚠ WARNING Failure to abide by the following precautions could lead to serious or possibly fatal injury. Always heed these precautions.

- Always turn off the power supply to the SRM1 before attempting any of the following.
  - Assembling any devices or racks.
  - Connecting or disconnecting any cables or wiring.

⚠ Caution Failure to abide by the following precautions could lead to faulty operation of the SRM1 or the system or could damage the SRM1 or CompoBus/S Units. Always heed these precautions.

- Fail-safe measures must be taken by the customer to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes.
- Construct a control circuit so that power supply for the I/O circuits does not come ON before power supply for the Unit. If power supply for the I/O circuits comes ON before power supply for the Unit, normal operation may be temporarily interrupted.
- If the operating mode is changed from RUN or MONITOR mode to PROGRAM mode, with the IOM Hold Bit ON, the output will hold the most recent status. In such a case, ensure that the external load does not exceed specifications. (If operation is stopped because of an operation error (including FALS instructions), the values in the internal memory of the CPU Unit will be saved, but the outputs will all turn OFF.)

- Use the Units only with the power supplies and voltages specified in the operation manuals.
- Take measures to stabilize the power supply to conform to the rated supply if it is not stable.
- Provide circuit breakers and other safety measures to provide protection against short-circuiting in external wiring.
- Install all Units according to instructions in the operation manuals.
- Do not install the Units in a place where they are subject to excessive noise in order to avoid any trouble or malfunction.
- Be sure that all the mounting screws, terminal screws, and cable connector screws are tightened to the torque specified in the relevant manuals. Incorrect tightening torque may result in malfunction.
- Double-check all the wiring before turning ON the power supply. Incorrect wiring may result in burning.
- Do not attempt to take any Units apart, to repair any Units, or to modify any Units in any way.
- Do not apply any impact to the Units.
- Use the cables specified in this manual and in reference manuals. Use crimp terminals when wiring the terminal block.
- Use a signal wire duct that is separate from the one used for high-tension lines or power lines.
- Be sure to confirm that the switch settings and wiring are correct before turning on the power supply.
- Check the user program for proper execution before actually running it on the Unit. Not checking the program may result in an unexpected operation.
- Confirm that the user programs run properly.
- Confirm that no adverse effect will occur in the system before attempting any of the following. Not doing so may result in an unexpected operation.
    - Changing the operating mode of the PC.
    - Force-setting/force-resetting any bit in memory.
    - Changing the present value of any word or any set value in memory.
- Before touching the Unit, be sure to first touch a grounded metallic object in order to discharge any static build-up. Not doing so may result in malfunction or damage.
- Use, store, and transport the Units within the specifications provided in this manual.
- Resume operation only after transferring to the new SRM1 the contents of the DM and HR Areas required for resuming operation. Not doing so may result in an unexpected operation.
- Do not pull on the cables or bend the cables beyond their natural limit. Doing either of these may break the cables.
- Do not place objects on top of the cables. Doing so may break the cables.
- When replacing parts, be sure to confirm that the rating of a new part is correct. Not doing so may result in malfunction or burning.
- Be sure to observe local ordinances and laws when disposing the Units.

⚠ Caution The following precautions are necessary to ensure the general safety of the system. Always heed these precautions.

- Provide double safety mechanisms to handle incorrect signals that can be generated by broken signal lines or momentary power interruptions.
- Provide external interlock circuits, limit circuits, and other safety circuits in addition to any provided within the SRM1 to ensure safety.

⚠ Caution Be sure to clear the memory before turning on the power supply to the delivered SRM1. The contents of the Data Memory (DM), Hold Relay (HR), and Counter (CNT) Areas in the CPU Unit may be cleared and the AR 1314 flag (which turns ON when the power interruption hold area is not held) may turn ON.

⚠ Caution Apply the SRM1 to a system that is not influenced by any undefined data even if the data in the DM, HR, or CNT area is cleared when the SRM1 has been turned off for a period exceeding the data backup period of the internal lithium battery. If the AR 1414 flag is ON, the data will be held unless it is turned OFF by the I/O Monitor, instructions, etc.
The system can be stopped by designating DM 6604 in the PC Setup so that a memory error occurs when the power interruption hold area is not held (with AR 1314 ON)

- A lithium battery in the CPU Unit is used to back up the counter values and the contents of the DM area, and HR area. The deterioration of the lithium battery capacity depends on the ambient temperature. The standard service life is 12 years under an ambient temperature of 40°C when operating 8 hours a day.

  If the power remains off for a period exceeding the data backup period, the contents of the Data Memory (DM), Hold Relay (HR), and Counter (CNT) Areas in the CPU Unit may be cleared and the AR 1314 flag (which turns ON when the power interruption hold area is not held) may turn ON.

  If the contents of the CPU Unit's program area are lost, the program stored in flash memory will be read to the CPU Unit's program area when the SRM1 is started up because the contents in the read-only area (DM 6144 through DM 6599) and PC Setup (DM 6600 through DM 6655) will be written to flash memory.
- However, if the power is turned off without changing the mode even if changes are made in the read-only DM area (DM 6144 through DM 6599), or PC Setup (DM 6600 through DM 6655) using a peripheral device, the contents of changes will not be written to flash memory. Although the data in these areas is backed up by the lithium battery, contents of changes will disappear if the service life of the lithium battery expires. In this case, programs in the flash memory will be automatically read into the user program memory.

  The changes can be saved by switching the SRM1 to RUN or MONITOR mode or turning off and restarting the SRM1 soon after the changes are made.

SECTION 1
Introduction

This section describes the SRM1's special features and functions and shows the possible system configurations.

1-1 SRM1 Features and Functions 2
1-1-1 Features 2
1-1-2 Functions 2
1-2 System Configuration 3
1-2-1 Basic Configuration 3
1-2-2 SRM1 Models 4
1-2-3 Peripheral Connections 5
1-3 Procedures From System Design to Test Operation 7
1-4 I/O and Data Area Allocations 8
1-4-1 I/O Allocations 8
1-4-2 Data Area Allocation 9

1-1 SRM1 Features and Functions

1-1-1 Features

The SRM1 is a special CompoBus/S controller that provides remote I/O with greatly reduced wiring. The SRM1 has no built-in I/O terminals, but it can provide the same I/O control as earlier PCs through the Slaves (Slave Terminals) that are used for I/O.

A decentralized I/O system with up to 32 Slaves can be constructed. The system can have up to 256 I/O points and these I/O points are controlled with the CompoBus/S System's high-speed response time of 1 ms max.

A very reliable and efficiently wired system can be constructed from special CompoBus/S components such as Analog Terminals (SRM1-C0□-V2 only), Remote Terminals, Sensor Terminals, Communications Cables, Connectors, and Terminators.

In SRM1-C0□-V2, the CompoBus/S system can be set to operate in long-distance communications mode in addition to the previous high-speed communications mode. This allows a main line length of up to 500 m so that I/O devices can be controlled from some distance away. The SRM1-C0□-V2 can also process analog data as well as digital I/O.

The SRM1's compact design allows for a smaller and thinner control panel.

The SRM1 is equipped with a program capacity of 4K words and a DM capacity of 2K words.

There are two SRM1(-V2) models available: the SRM1-C02-V2, which is equipped with an RS-232C port and communications functions, and the very cost-effective SRM1-C01-V2, which is not equipped with an RS-232C port.

1-1-2 Functions

Interval Timer Function

The SRM1 is equipped with an interval timer which can be set from 0.5 ms to 319,968 ms in units of 0.1 ms. The timer can be set to trigger a single interrupt (one-shot mode) or repeat scheduled interrupts (scheduled interrupt mode).

(The interrupts pause execution of the main program while an interrupt program is executed.)

Low-maintenance Design

Memory can be backed up without a battery by using flash memory.

Communications

The SRM1 can communicate with PCs or other devices via Host Link, 1:1 NT Link, 1:N NT Link, 1:1 PC Link, or RS-232C communications.

Port	Applicable communications functions
Peripheral Port	Peripheral device connections, Host Link, and RS-232C communications
RS-232C Port	Host Link, 1:1 NT Link, 1:N NT Link, 1:1 PC Link, and no-protocol (RS-232C) communications

Programming Using the PT

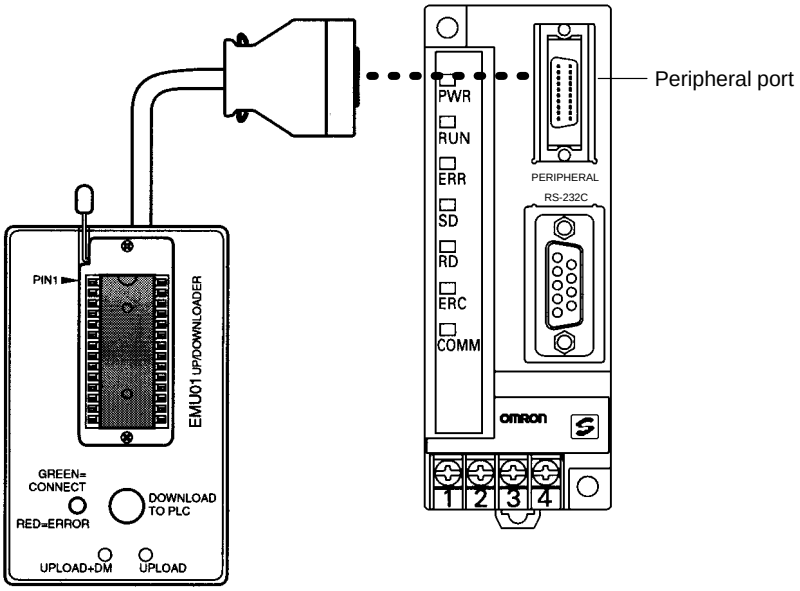
Programming is possible through the PT (Programmable Terminal) screen using an OMRON PT that contains Programming Console functions. (This applies only to the SRM1-C02-V1 and SRM1-C02-V2.)

Standard Peripheral Devices

The SRM1 uses the same Programming Consoles, CX-Programm*er*, SYSMAC-CPT, and SYSMAC Support Software (SSS) as the Mini H-type, CQM1, CPM1/CPM1A, and CPM2A/CPM2C PCs.

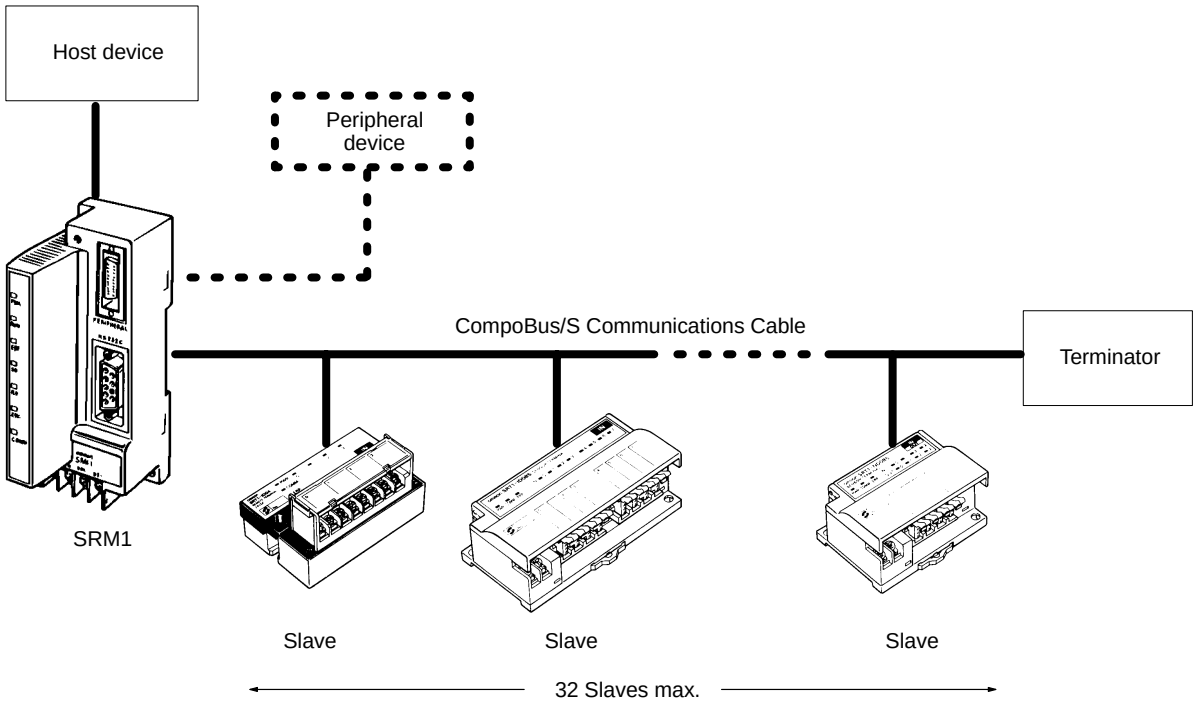
Expansion Memory Unit

The CPM1-EMU01-V1 Expansion Memory Unit is a program loader for small-size or micro PCs. Using the CPM1-EMU01-V1, simple on-site transfer of user programs and data memory is possible with PCs.



1-2 System Configuration

1-2-1 Basic Configuration



1-2-2 SRM1 Models

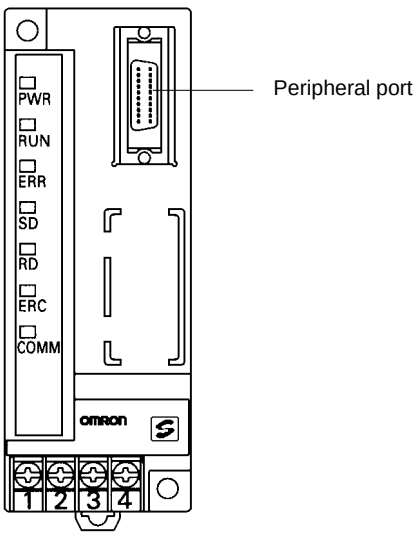
Model	RS-232C port	PT programming functions
SRM1-C01-V2	No	No
SRM1-C02-V2	Yes	Yes

The following table compares the functions in the SRM1(-V2) PCs with the functions in earlier SRM1 PCs.

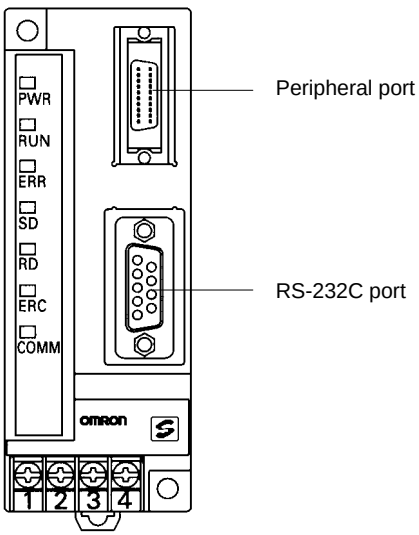
Function	SRM1 models		
	SRM1-C0□-V2	SRM1-C0□-V1	SRM1-C0□
Data backup	Backed up by a lithium battery with a minimum lifetime of 10 years at 25°C.		Capacitor backup
Programming Console functions	Programming can be performed through a Programming Console connected to the peripheral port or an OMRON PT connected to the RS-232C port.		Programming can be performed through a Programming Console connected to the peripheral port.
Data processing	Bit data (ON/OFF for 16 bits) and 16-bit analog data from Analog Units	Bit data (ON/OFF status of bits)	
Communications	High-speed communications (previous mode) or long-distance communications mode	High-speed communications mode (previous mode) only.	
Connections with host devices	Host Link, no-protocol, 1:1 NT Link, 1:N NT Link, and 1:1 PC Link communications	Host Link, no-protocol, 1:1 NT Link, and 1:1 PC Link communications	
Instructions	The instructions in earlier SRM1 PCs plus the following instructions: NEG(—), PID(—), SCL(66), and ZCP(—)	Basic instructions: 14 Special instructions: 77 (123 variations)	

Note The Analog Terminal can be used as a slave only with version-2 models. Incorrect data may be transferred if an Analog Terminal is used with the wrong model.

SRM1-C01-V2
(No RS-232C port)



SRM1-C02-V2
(With RS-232C port)



1-2-3 Peripheral Connections

The following peripherals can be connected to the SRM1(-V2) PCs. Refer to *Appendix A Standard Models* for a complete list of connectable peripherals.

Slaves

The following table shows the Slaves that can be connected. Refer to the *CompoBus/S Operation Manual* (W266) for more details.

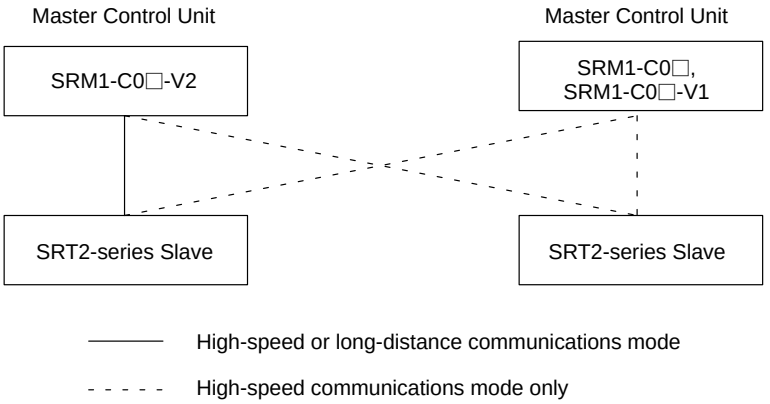
Slave	SRT2 Series High-speed or long-distance communications	SRT1 Series High-speed communications only
Remote Terminals (transistors)	SRT2-ID04	SRT1-ID04
	SRT2-ID04-1	SRT1-ID04-1
	SRT2-ID08	SRT1-ID08
	SRT2-ID08-1	SRT1-ID08-1
	SRT2-ID16	SRT1-ID16
	SRT2-ID16-1	SRT1-ID16-1
	SRT2-ID16T	None
	SRT2-ID16T-1	
	SRT2-OD04	SRT1-OD04
	SRT2-OD04-1	SRT1-OD04-1
	SRT2-OD08	SRT1-OD08
	SRT2-OD08-1	SRT1-OD08-1
	SRT2-OD16	SRT1-OD16
	SRT2-OD16-1	SRT1-OD16-1
	SRT2-OD16T	None
	SRT2-OD16T-1	
	SRT2-MD16T	
	SRT2-MD16T-1	
Connector Terminals (transistors)	SRT2-VID08S	None
	SRT2-VID08S-1	
	SRT2-VID16ML	
	SRT2-VID16ML-1	
	SRT2-VOD08S	
	SRT2-VOD08S-1	
	SRT2-VOD16ML	
	SRT2-VOD16ML-1	
Remote Terminals (relays)	SRT2-ROC08	SRT1-ROC08
	SRT2-ROC16	SRT1-ROC16
Remote Terminals (power MOSFET)	SRT2-ROF08	SRT1-ROF08
	SRT2-ROF16	SRT1-ROF16
Remote Modules	None	SRT1-ROF08
		SRT1-ROF16
Analog Input Terminal	SRT2-AD04	None
Analog Output Terminal	SRT2-DA02	
Sensor Amplifier Terminals	SRT2-TID04S (See note 3.)	SRT1-TID04S
	SRT2-TKD04S (See note 3.)	SRT1-TKD04S
Sensor Terminals	SRT2-ID08S (See note 3.)	SRT1-ID08S
	SRT2-OD08S (See note 3.)	SRT1-OD08S
	SRT2-MD08S (See note 3.)	SRT1-MD08S

Slave	SRT2 Series High-speed or long-distance communications	SRT1 Series High-speed communications only
Bit Chain Terminal	None	SRT1-B1T
I/O Link Unit	CPM1A-SRT21	None

Note
1. SRT1-series Remote Terminals and Sensor Terminals can operate in high-speed communications mode only. Be sure to use SRT2-series Remote Terminals and Sensor Terminals when the SRM1-C0□-V2 is used in long-distance communications mode.
2. The Analog I/O Terminals can be used with SRM1-C0□-V2 only.
3. To be marketed in the near future.

Compatible Communications Modes

The long-distance communications mode can be used between SRM1-C0□-V2 Master Control Units and SRT2-series Slaves only, as shown in the following diagram. High-speed mode must be used if even one SRT1-series Slave is included in the CompoBus/S system.



Peripheral Devices

The SRM1 can use a Programming Console or a personal computer running CX-Programmer, SYSMAC Support Software (SSS), or SYSMAC-CPT as a Peripheral Device.

Programming Console

Programming Consoles are compact Peripheral Devices that support basic functions such as writing ladder programs and monitoring SRM1 operation. They are useful for onsite operations.

Refer to *Section 4 Using a Programming Console* for details on Programming Console operations.

SYSMAC Support Software (SSS), SYSMAC-CPT, and CX-Programmer

In addition to the basic Programming Console operations, the CX-Programmer, SYSMAC-CPT, and SSS can be used to edit ladder programs offline, save programs to disk, and perform high-level monitoring; the CX-Programmer, SYSMAC-CPT and SSS functions allow the user to design more efficient ladder programs. Refer to the manuals listed on the *About this Manual* page for details on using them.

SYSMAC Support Software (SSS)

When using the SSS, set the PC model to the “CQM1” and observe the following restrictions.

- Addresses will be checked according to the CQM1 address ranges, which are wider than the SRM1 address ranges. Be sure to use only the allowable addresses.
- The CQM1 has a larger memory than the SRM1 and the amount of memory available display will not be correct. Allow for the difference between capacities.

- Instructions will be displayed that are not supported by the SRM1. Do not use these instructions.

CX-Programmer and SYSMAC-CPT

The following instructions cannot be programmed when using the CX-Programmer or SYSMAC-CPT. Errors will occur if an attempt is made to transfer them from the PC to the computer: SCL(66)/@SCL(66), ZCP, NEG/@NEG, and PID.

Host Computers, PTs, and PCs

An SRM1(-V2) can be connected an IBM PC/AT or compatible computer or OMRON PT through an RS-232C Adapter (CPM1-CIF01) mounted to the SRM1's peripheral port. (Use Host Link mode when connecting a PT through an RS-232C Adapter.)

The RS-232C port on an SRM1-C02-V2 can be used to connect directly to an IBM PC/AT or compatible computer, OMRON PT, or PC (C200HX/HG/HE, C200HS, CQM1, CPM1, CPM1A, CPM2A, or CPM2C). (Use Host Link mode or NT Link mode when connecting a PT directly.)

1-3 Procedures From System Design to Test Operation

The procedures from system design to test operation are explained in the sections of this manual as follows and in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual* (W353):

1, 2, 3...

1. **System Design**
   Refer to *3-1 System Design*.
2. **Installation**
   Refer to *3-3 Installing the SRM1*.
3. **Wiring**
   Refer to *3-4 Wiring and Connections*.
4. **Creating the Ladder Program**
   Refer to the relevant sections in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual (W353)* and the applicable PC manual.
5. **Inputting the Program**
   Refer to *Section 4 Using the Programming Console*, *Section 5 Test Runs and Error Processing*, *CX-Programmer Users Manual* (W346), *SYSMAC Support Software (SSS) Operation Manuals* (W247 and W248), and *SYSMAC-CPT Support Software Quick Start Guide* (W332) and *User Manual* (W333).
6. **Test Operation**
   Refer to *5-1-2 SRM1 Test Run Procedure*.

1-4 I/O and Data Area Allocations

1-4-1 I/O Allocations

The input bits of SRM1 words 000 to 007, and the output bits of words 010 to 017, are allocated to the CompoBus/S Slave. These allocations are shown in the following table.

I/O	Word address	Bits	
		15 to 08	07 to 00
Inputs	000	IN1	IN0
	001	IN3	IN2
	002	IN5	IN4
	003	IN7	IN6
	004	IN9	IN8
	005	IN11	IN10
	006	IN13	IN12
	007	IN15	IN14
Outputs	010	OUT1	OUT0
	011	OUT3	OUT2
	012	OUT5	OUT4
	013	OUT7	OUT6
	014	OUT9	OUT8
	015	OUT11	OUT10
	016	OUT13	OUT12
	017	OUT15	OUT14

IN0 to IN15 are Input Slave node numbers, and OUT0 to OUT15 are Output Slave node numbers.

If the maximum number of CompoBus/S devices is set to 16, then IN8 to IN15 and OUT8 to OUT15 can be used as work bits.

Words IR 008, IR 009, IR 018, and IR 019 can be used as work words.

The bits for two node number are allocated to 16-point Slaves so that all bits are in the same word. If an even node address is set, the node address that is set and the next node address following it will be used. For example, if node address 6 is set for a 16-point Output Slave, bits for node addresses OUT6 and OUT7 will be used. If an odd node address is set, the node address that is set and the previous node address will be used. For example, if node address 3 is set for a 16-point Output Slave, bits for node addresses OUT2 and OUT3 will be used.

All of the bits for one node address are allocated to a 4-point Slave. If an even numbered node address is set, bits 00 to 03 are used and bits 04 to 07 are not used. If an odd numbered node address is set, bits 8 to 11 are used and bits 12 to 15 are not used.

Analog Terminals are allocated from 16 to 64 bits per Terminals as shown in the following table. If an allocation is not completely within the input or output area, communications will not be possible and the COMM indicator will not be lit.

I/O bits allocated	Node address setting	Node addresses used	Address setting range
64 bits (SRT2-AD04, 4 analog inputs)	Even	Set address to set address + 7	0 to 9
	Odd	Set address – 1 to set address + 6	
48 bits (SRT2-AD04, 3 analog inputs)	Even	Set address to set address + 5	0 to 11
	Odd	Set address – 1 to set address + 4	
32 bits (SRT2-AD04, 2 analog inputs) (SRT2-DA02, 2 analog outputs)	Even	Set address to set address + 3	0 to 13
	Odd	Set address – 1 to set address + 2	
16 bits (SRT2-AD04, 1 analog input) (SRT2-DA02, 1 analog output)	Even	Set address to set address + 1	0 to 15
	Odd	Set address – 1 to set address	

Examples
If node address 3 is set for the SRT2-DA02 and 2 analog outputs are used, 32 bits are allocated from OUT2 to OUT5.

If node address 10 is set for the SRT2-AD04 and 4 analog inputs are used, the allocated area would exceed the output area available for allocation and communications would not be possible.

1-4-2 Data Area Allocation

The relationships between the data areas and words that can be used by the SRM1 are shown in the following table. For details, refer to the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual* (W353).

Name	Number of words or bits	Word addresses
Input bits	8 words	IR 000 to IR 007
Output bits	8 words	IR 010 to IR 017
Work bits	44 words (See note 1.)	IR 008 and IR 009, IR 018 and IR 019, IR 200 to IR 239
SR area	16 words	IR 240 to IR 255
HR area	20 words	HR 00 to HR 19
AR area	16 words	AR 10 to AR 15 (See note 2.)
LR area	16 words	LR 00 to LR 15
DM area (Read/Write)	2,022 words	DM 0000 to DM 2021
DM area (Read Only)	456 words	DM 6144 to DM 6599
DM area (PC Setup)	56 words	DM 6600 to DM 6655
TR area	8 bits	TR 0 to TR 7
TIM/CNT area	128 bits	TIM/CNT 000 to 127

Note 1. When the CompoBus/S system is used in 128-bit mode, IR 004 to IR 007 and IR 014 to IR 017 can be used as work words.

2. AR 04 to AR 07 are used for Slave status.

SECTION 2
Specifications and Components

This section provides the technical specifications of the SRM1(-V2) and describes its main components.

2-1 Specifications 12
2-1-1 General Specifications 12
2-1-2 Characteristics 13
2-1-3 CompoBus/S Communications Specifications 14
2-2 Unit Components 15

2-1 Specifications

2-1-1 General Specifications

Item	SRM1-C01/C02-V2
Supply voltage	24 VDC
Allowable supply voltage	20.4 to 26.4 VDC
Power consumption	3.5 W max.
Inrush current	5.0 A max. (pulse width: 15 ms max.)
Noise immunity	Conforms to IEC61000-4-4; 2 kV (power lines)
Vibration resistance	10 to 57 Hz, 0.075-mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s^2 in X, Y, and Z directions for 80 minutes each (Time coefficient; 8 minutes × coefficient factor 10 = total time 80 minutes)
Shock resistance	147 m/s^2 three times each in X, Y, and Z directions
Ambient temperature	Operating: 0°C to 55°C Storage: –20°C to 75°C
Absolute humidity	10% to 90% (with no condensation)
Atmosphere	Must be free from corrosive gas.
Terminal screw size	M3
Power interrupt time	DC type: 2 ms min.
Weight	150 g max.

2-1-2 Characteristics

Item	SRM1-C01/C02-V2
Control method	Stored program method
I/O control method	Cyclic scan method
Programming language	Ladder diagram
Instruction length	1 step per instruction, 1 to 5 words per instruction
Types of instructions	Basic instructions: 14 Special instructions: 81 instructions, 125 variations
Execution time	Basic instructions: 0.97 μs (LD instruction) Special instructions: 9.1 μs (MOV instruction)
Program capacity	4,096 words
Maximum number of I/O points	256 points
Input bits	00000 to 00715 (Words not used as input words can be used as work words.)
Output bits	01000 to 01715 (Bits not used as output bits can be used as work bits.)
Work bits	704 bits: 00800 to 00915 (Words IR 008 and IR 009) 01800 to 01915 (Words IR 018 and IR 019) 20000 to 23915 (Words IR 200 to IR 239)
Special bits (SR area)	248 bits: 24000 to 25507 (Words IR 240 to IR 255)
Temporary bits (TR area)	8 bits (TR0 to TR7)
Holding bits (HR area)	320 bits: HR 0000 to HR 1915 (Words HR 00 to HR 19)
Auxiliary bits (AR area)	256 bits: AR 0000 to AR 1515 (Words AR 00 to AR 15)
Link bits (LR area)	256 bits: LR 0000 to LR 1515 (Words LR 00 to LR 15)
Timers/Counters	128 timers/counters (TIM/CNT 000 to TIM/CNT 127) 100-ms timers: TIM 000 to TIM 127 10-ms timers (high-speed counter): TIM 000 to TIM 003 Decrementing counters and reversible counters (Note: TIMH(15) will not time reliably if the cycle time is over 10 ms and timer numbers TIM 004 to TIM 127 are used.)
Data memory	Read/Write: 2,022 words (DM 0000 to DM 2021) Read-only: 512 words (DM 6144 to DM 6655)
Interval timer interrupts	One-shot mode/Scheduled interrupt mode, one bit (0.5 to 319,968 ms)
Memory protection	HR, AR, and DM area contents; and counter values maintained during power interruptions.
Memory backup	Flash memory: The program and read-only DM area are backed up without a battery. Lithium battery backup: The read/write DM area, HR area, AR area, and counter values are backed up by the lithium battery whose service life extends over ten years under an ambient temperature of 25°C. (Note: The lifetime of the lithium battery capacity depends on the ambient temperature. Refer to the descriptions on the next page.)
Self-diagnostic functions	CPU Unit failure (watchdog timer), memory check, communications errors, setting errors
Program checks	No END instruction, programming errors (continuously checked during operation)
Peripheral port	One point; tool connection, Host Link, no protocol
RS-232C Port	One point (SRM1-C02-V2 only); Host Link, 1:1 NT Link, 1:N NT Link, 1:1 PC Link, no protocol

Backup Time vs. Temperature A lithium battery in the CPU Unit is used to back up the contents in the user program area, the READ/WRITE area in the Data Memory (DM), Hold Relay (HR), the Auxiliary Memory Relay (AR), and in the data area of the Counter (CNT). The deterioration of the lithium battery capacity depends on the ambient temperature. The standard service life is 12 years under an ambient temperature of 40°C when operating 8 hours a day.

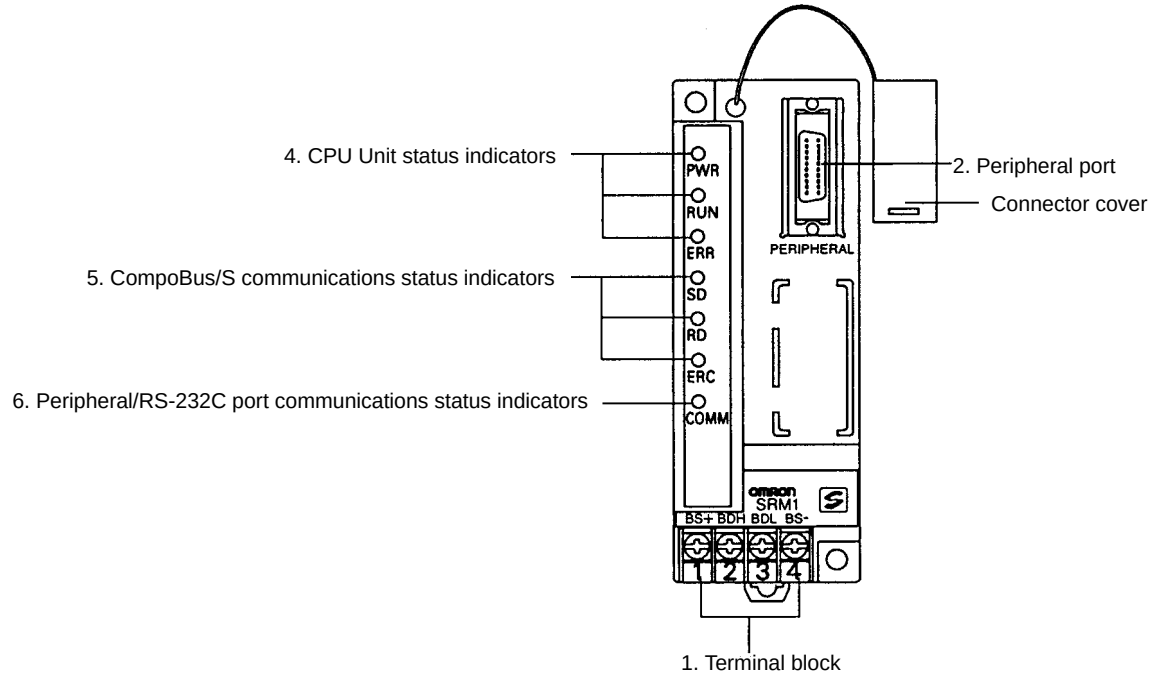
If the power supply is interrupted after the lithium battery capacity has deteriorated, the contents in the user program area, the READ/WRITE area in the Data Memory (DM), Hold Relay (HR), Auxiliary Memory Relay (AR), and in the data area of the Counter (CNT) may be lost. Even if the contents of the CPU Unit's program area are lost, however, the user program and DM read-only contents (including the one in the PC Setup area) stored in flash memory will be read to the CPU Unit's user program area when the SRM1 is next started up.

2-1-3 CompoBus/S Communications Specifications

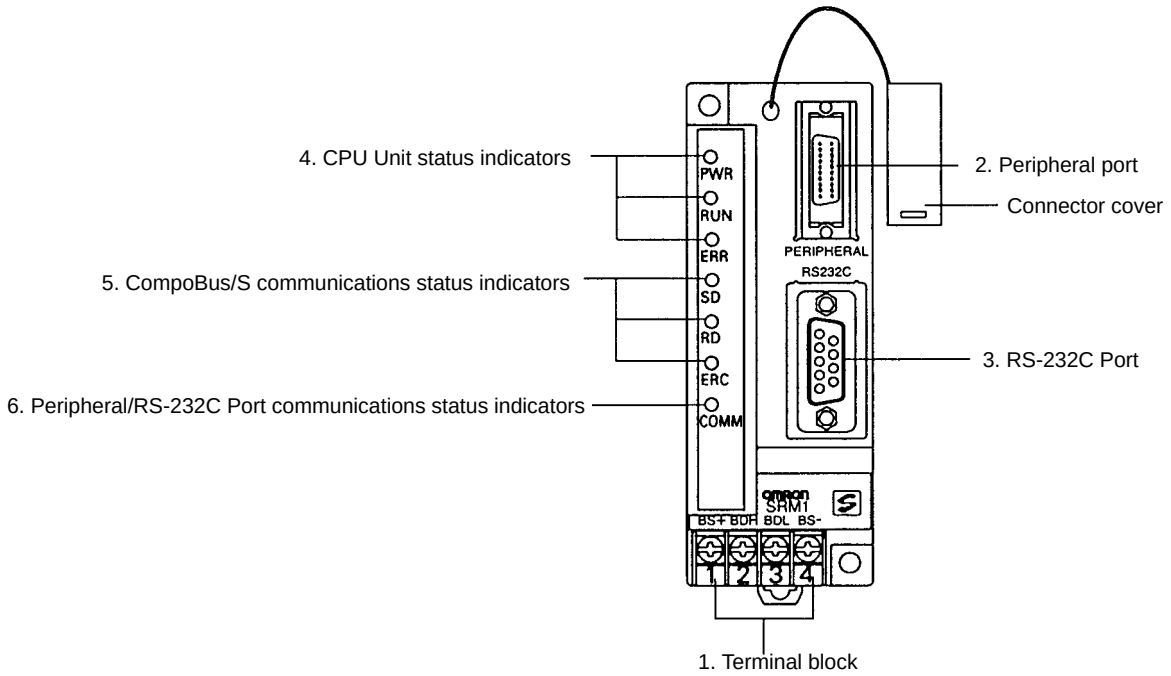
Item		Specifications
Communications method		CompoBus special protocol
Transmission method		Multi-drop, T-branch
Baud rate	High-speed communications mode	750 kbps
	Long-distance communications mode	93.75 kbps
Modulation method		Baseband method
Code method		Manchester coding method
Maximum number of connectible terminals		32: 16 IN and 16 OUT
		16: 8 IN and 8 OUT
Number of points per frame		256 (128 IN and 128 OUT), when maximum number of connectible terminals is 32.
		128 (64 IN and 64 OUT), when maximum number of connectible terminals is 16.
Communications cycle time	High-speed communications mode	0.8 ms, when maximum number of terminals is set to 32.
		0.5 ms, when maximum number of terminals is set to 16.
	Long-distance communications mode	6.0 ms, when maximum number of terminals is set to 32.
		4.0 ms, when maximum number of terminals is set to 16.
Communications function		Cyclic transfer only (no message communications)
Error control checks		Manchester code check, frame length check, parity check, two-transfer comparison
Communications distance	High-speed communications mode	Main line length: 100 m max. Branch line length: 3 m max. Total branch line length: 50 m max.
	Long-distance communications mode	Main line length: 500 m max. Branch line length: 6 m max. Total branch line length: 120 m max.
Cable	Vinyl-clad VCTF JIS C 3306	Two 0.75 mm^2 conductors (2 signal wires)
	Flat cable	Four 0.75 mm^2 conductors (2 signal wires and 2 power supply wires)

2-2 Unit Components

SRM1-C01-V2



SRM1-C02-V2



1) Terminal Block These terminals connect the power supply (24 V) and the CompoBus/S transmission path. For details regarding power supply wiring, refer to *3-4-2 Power Supply Wiring*.

2) Peripheral Port The Peripheral Port connects the programming tool or an RS-232C or RS-422 adapter. Be sure to use the correct cable.

3) RS-232C Port The RS-232C Port connects to an RS-232C interface such as a personal computer or an OMRON PT. For details, refer to *3-4-4 RS-232C Port Wiring*.

4, 5, 6) Indicators

There are three types of LED indicators: CPU Unit status indicators, CompoBus/S communications status indicators, and peripheral/RS-232C Port communications status indicators. These indicate the status of various Units, as shown in the following table.

Indicator	Display	Status
PWR (Green)	ON	Power is being supplied.
	OFF	Power is not being supplied.
RUN (Green)	ON	In RUN mode or MONITOR mode
	OFF	In PROGRAM mode or fatal error has occurred.
ERR (Red)	ON	Fatal error has occurred.
	Flashing	Non-fatal error has occurred.
	OFF	Normal operation
SD (Yellow)	ON	CompoBus/S data is being sent.
	OFF	Data is not being sent.
RD (Yellow)	ON	CompoBus/S data is being received.
	OFF	Data is not being received.
ERC (Red)	ON	A CompoBus/S communications error has occurred.
	OFF	Normal operation
COMM (Yellow)	Flashing	Data is being sent or received at the Peripheral Port or RS-232C Port.
	OFF	Data is not being sent or received.

SECTION 3
Installation and Wiring

This section explains how to install and wire the SRM1(-V2). Be sure to follow the instructions contained here concerning the control panel, power supply, CompoBus/S transmissions, and RS-232C Port wiring. For details regarding the wiring of CompoBus/S Terminal transmission paths and I/O, refer to the *CompoBus/S Operation Manual (W266)*.

3-1 System Design 18
3-1-1 Power Supply Wiring 18
3-1-2 Interlock and Limit Circuits 18
3-1-3 Power Supply Sequence 18
3-2 Selecting an Installation Site 19
3-2-1 Installation Site Conditions 19
3-2-2 Panel/Cabinet Installation 19
3-3 Installing the SRM1 20
3-3-1 Surface Installation 20
3-3-2 DIN Track Installation 20
3-4 Wiring and Connections 21
3-4-1 General Precautions for Wiring 21
3-4-2 Power Supply Wiring 23
3-4-3 CompoBus/S Transmission Line Wiring 23
3-4-4 RS-232C Port Wiring 24
3-4-5 Host Link Connections 26
3-5 One-to-one NT Link 30
3-6 One-to-N NT Link 30
3-7 One-to-one PC Link Connections 31
3-7-1 Basics 31
3-7-2 Restrictions 32
3-7-3 Cable Connections 32
3-7-4 PC Setup Settings 32

3-1 System Design

Take the points covered in this section into consideration when designing the system.

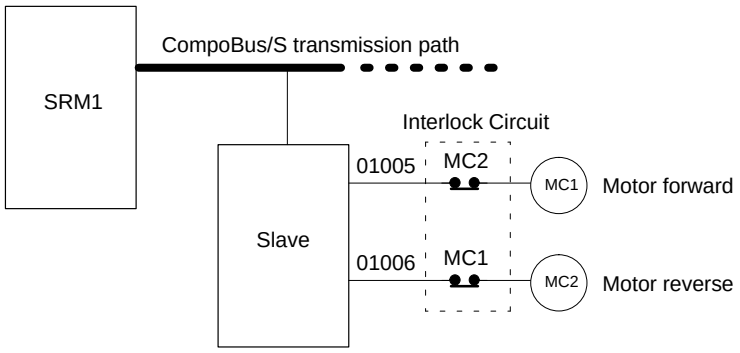
3-1-1 Power Supply Wiring

Separate the power supply wiring from the control system, SRM1 system, and DC I/O system wiring.

3-1-2 Interlock and Limit Circuits

Construct an external interlock circuit if SRM1 outputs are used to perform reciprocal operations such as controlling the forward and reverse operation of a motor or if incorrect SRM1 operation could cause accidents or mechanical damage. Also, construct an external limit circuit to prevent run-away movement in operations such as position control.

The following diagram shows an example of an interlock circuit.



In the interlock circuit above, MC1 and MC2 cannot be ON at the same time even if SRM1 outputs 01005 and 01006 are both ON at the same time (an incorrect operation).

3-1-3 Power Supply Sequence

Time Up to the Start of Operation

The time from when the power supply is turned on to when the operation starts varies depending on the operation conditions such as power supply voltage, configuration, ambient temperature, etc. The minimum time is approximately 500 ms and the maximum is approximately 1.1 s.

Momentary Power Failure Detection

A momentary power failure (i.e., a voltage drop to less than 85% of the rated voltage) lasting less than 2 ms is not detected and the SRM1 continues to operate.

A momentary power failure lasting longer than 2 ms may cause the SRM1 to stop operation. If this occurs, operation will be automatically resumed when the rated voltage again rises above 85%.

Note The SRM1 may repeat stop/start operations if the supply voltage of less than 85% of the rated value gradually goes up or down. If this affects the equipment, etc., provide a protection circuit which shuts off the output if the supply voltage is not above the rated value.

The output status of Slaves when the SRM1 is stopped can be set on the Slave side either to have the ON/OFF status directly prior to the stop retained or to have all outputs turned OFF.

3-2 Selecting an Installation Site

The SRM1 is resistant to harsh conditions and highly reliable, but installing it in a favorable site will maximize its reliability and operating lifetime.

3-2-1 Installation Site Conditions

Avoid installing the SRM1 in a site with any of the following conditions.

- Where the SRM1 is exposed to direct sunlight.
- Where the ambient temperature is below 0°C or over 55°C.
- Where the SRM1 may be affected by condensation due to radical temperature changes.
- Where the ambient humidity is below 10% or over 90%.
- Where there is any corrosive or inflammable gas.
- Where there is excessive dust, saline air, or metal powder.
- Where the SRM1 is affected by vibration or shock.
- Where any water, oil, or chemical may splash on the SRM1.

Be sure that the conditions at the installation site conform to the SRM1's general specifications. Refer to *2-1-1 General Specifications* for details.

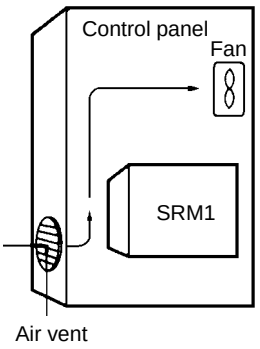
3-2-2 Panel/Cabinet Installation

Consider PC operation, maintenance, and surrounding conditions when installing the SRM1 in a panel or cabinet.

Overheating

The operating temperature range for the SRM1 is 0°C to 55°C. Be sure that there is adequate ventilation for cooling.

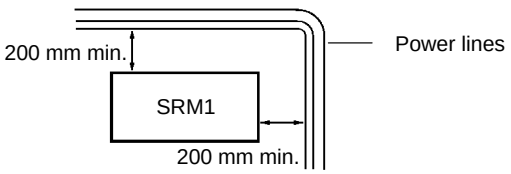
- Allow enough space for air circulation.
- Do not install the SRM1 above equipment that generates a large amount of heat, such as heaters, transformers, or large resistors.
- Install a cooling fan or system when the ambient temperature exceeds 55°C.



Electrical Noise

Power lines and high-voltage equipment can cause electrical noise in the PC.

- Do not install the SRM1 in a panel or cabinet with high-voltage equipment.
- Allow at least 200 mm between the SRM1 and nearby power lines.



Accessibility

Ensure that the SRM1 can be accessed for normal operation and maintenance.

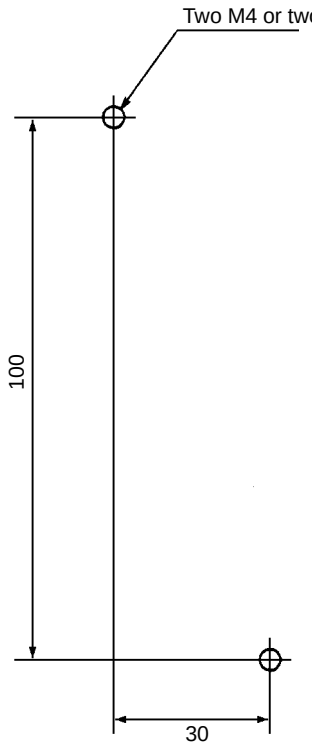
- Provide a clear path to the SRM1 for operation and maintenance. High-voltage equipment or power lines could be dangerous if they are in the way during routine operations.
- The SRM1 will be easiest to access if the panel or cabinet is installed about 3 to 5 feet off of the floor.

3-3 Installing the SRM1

The SRM1 can be installed on a horizontal surface or on a DIN track.

3-3-1 Surface Installation

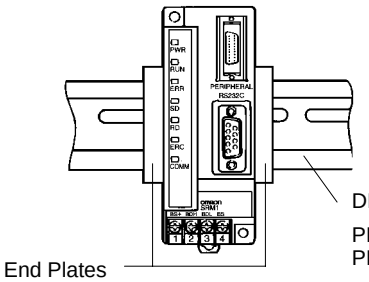
Use the following pattern when installing an SRM1 on a horizontal surface.



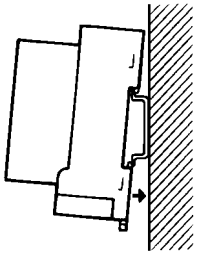
(Unit: mm, with tolerance of ±0.2 mm)

3-3-2 DIN Track Installation

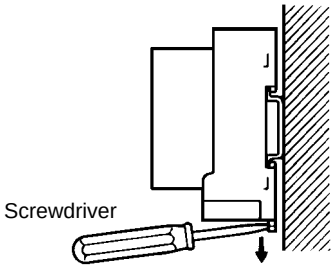
The SRM1 can be installed on a 35-mm DIN track.



Installation
Lower the SRM1 so that the notch on the back of the Unit catches the top of the DIN Track. Push the Unit forward until the lock snaps into place.



Removal
Pry the lock down with a standard screwdriver and pivot the Unit upward to remove it.



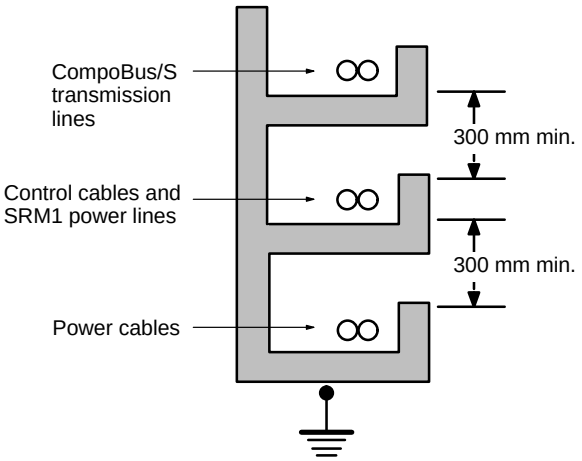
3-4 Wiring and Connections

3-4-1 General Precautions for Wiring

I/O Line Noise

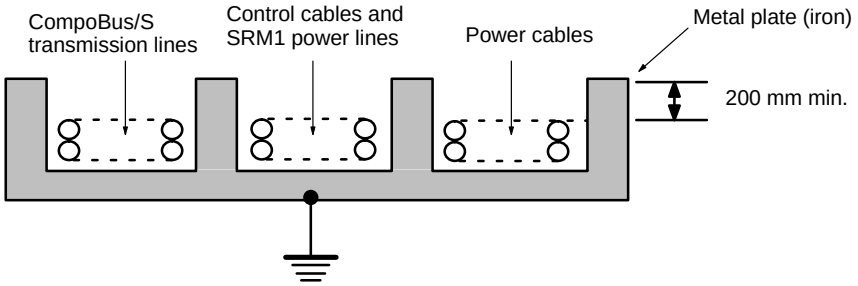
To avoid noise, do not run CompoBus/S transmission lines in the same duct or conduit as power lines.

Hanging Ducts
Leave at least 300 mm between the ducts, as shown in the following diagram.



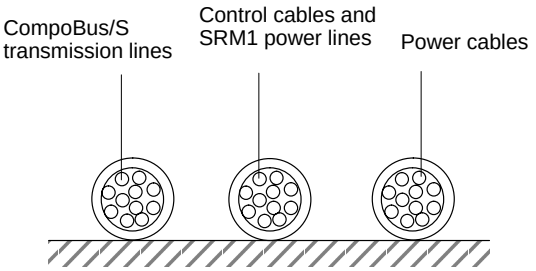
Floor Ducts

Leave at least 200 mm between the wiring and the top of the duct, as shown in the following diagram.



Conduit

Separate the CompoBus/S transmission lines, power and control lines, and power cables, as shown in the following diagram.

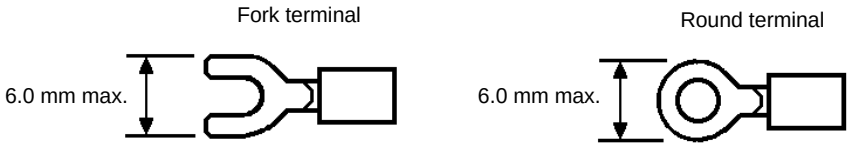


Terminal Connections

Always use crimp connectors for the SRM1's power lines and transmission lines.

Use M3 terminal screws and tighten the screws securely (to a torque of 0.48 N • m).

Recommended Terminals: Use the terminals shown below.



3-4-2 Power Supply Wiring

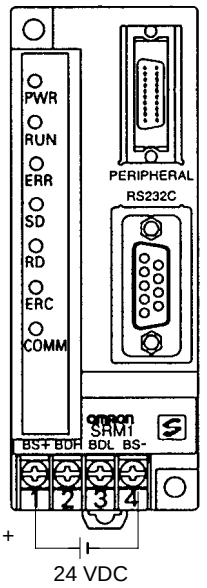
Applicable Power Supply

Use a power supply that conforms to specifications of at least 24 VDC and 3.5 W. OMRON's S82K-00724 is recommended (input: 100 VAC; output: 24 VDC, 7.5 W).

Note The above power supply is for a case where the CompoBus/S Slave is separated from the power supply. Refer to the *CompoBus/S Operation Manual (W266)* when supplying power to the whole system including the Slave.

Wiring Connections

Wire the power supply as shown in the following diagram.



Note Be sure to ground the ground terminal of the power supply.

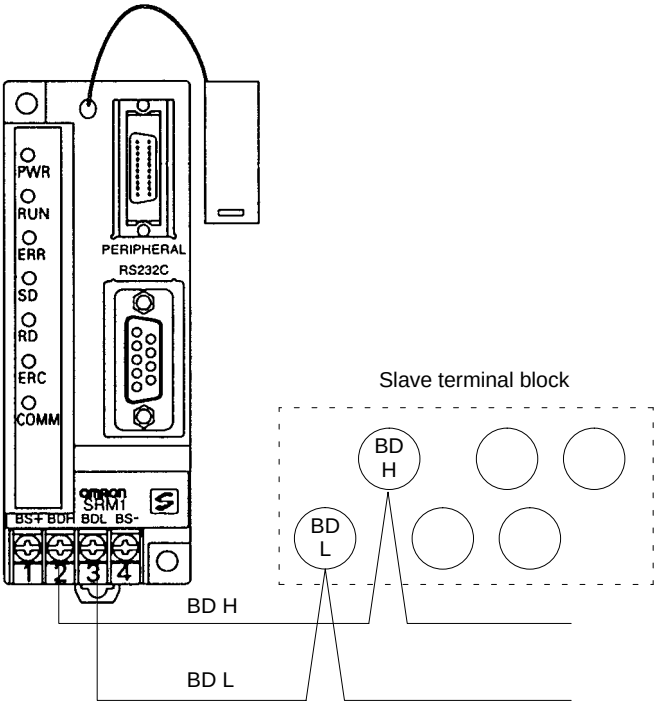
3-4-3 CompoBus/S Transmission Line Wiring

Applicable Cable

Be sure to use the specified cable, and do not mix flat cable and VCTF cable.

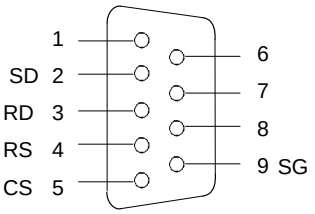
Cable	Model number	Specifications
Special-purpose flat cable	XBIT-W10	4-conductor flat cable: 0.75 mm^2
VCTF cable	---	2-conductor Vinyl-clad VCTF JIS C3306 VCTF 0.75x2C

Wiring Connections Wire the CompoBus/S transmission lines as shown in the following diagram.



3-4-4 RS-232C Port Wiring

Connector Pin Arrangement The following diagram shows the connector pin arrangement for the RS-232C port, i.e., the SRM1 (SRM1-C02-V2) and RS-232C Adapter (CPM1-CIF01).



Cable Connections The following diagrams show the communications cable connections between the RS-232C port, i.e., the SRM1 (SRM1-C02-V2) and RS-232C Adapter (CPM1-CIF01), and the various external devices.

IBM PC/AT or Compatible Computer

Signal	Pin No.
CD	1
RD	2
SD	3
ER	4
SG	5
DR	6
RS	7
CS	8
CI	9

RS-232C Port

Pin No.	Signal
1	–
2	SD
3	RD
4	RS
5	CS
6	–
7	–
8	–
9	SG
Hood	

OMRON PT or PC*

Signal	Pin No.
–	1
SD	2
RD	3
RS	4
CS	5
–	6
–	7
–	8
SG	9

RS-232C Port

Pin No.	Signal
1	–
2	SD
3	RD
4	RS
5	CS
6	–
7	–
8	–
9	SG
Hood	

(Recommended Cables)

XW2Z-200T: 2 m
XW2Z-500T: 5 m

*Host Link or NT Link with an OMRON PT, or 1:1 PC Link with a SYSMAC C200HX/HE/HG/HS, CQM1, or CPM1 Programmable Controller.

3G2A9-AL004-E Link Adapter

1:1 Connection

RS-232C Port

Signal	Pin No.
FG	1
SD	2
RD	3
RS	4
CS	5
DR	6
SG	7
–	8
–	9
ER	20

Pin No.	Signal
1	–
2	SD
3	RD
4	RS
5	CS
6	–
7	–
8	–
9	SG
Hood	

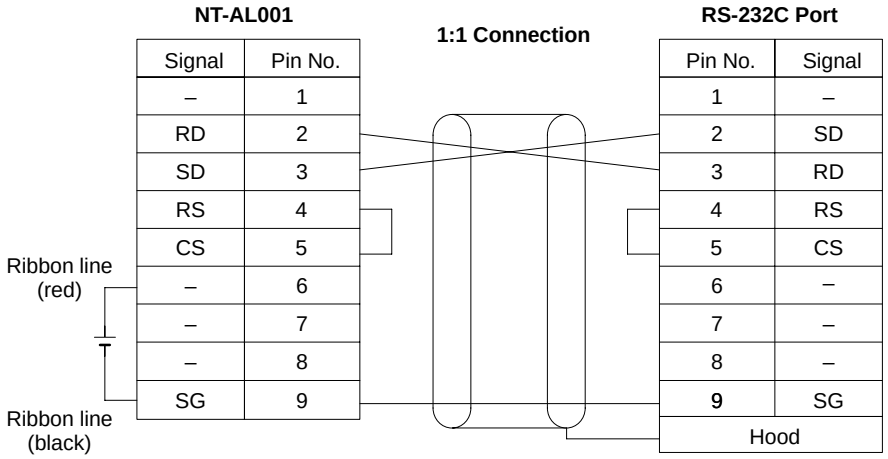
3G2A9-AL004-E Link Adapter

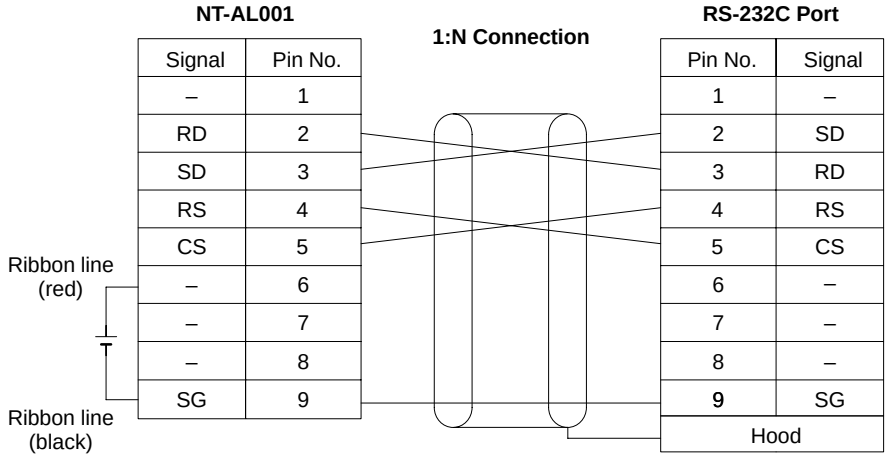
1:N Connection

RS-232C Port

Signal	Pin No.
FG	1
SD	2
RD	3
RS	4
CS	5
DR	6
SG	7
–	8
–	9
ER	20

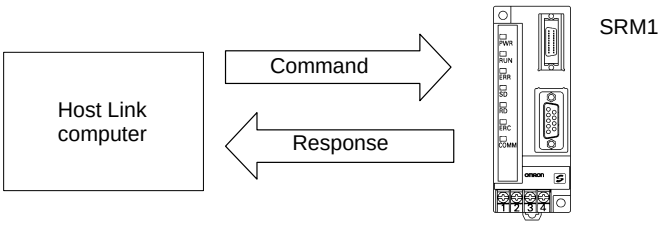
Pin No.	Signal
1	–
2	SD
3	RD
4	RS
5	CS
6	–
7	–
8	–
9	SG
Hood	





3-4-5 Host Link Connections

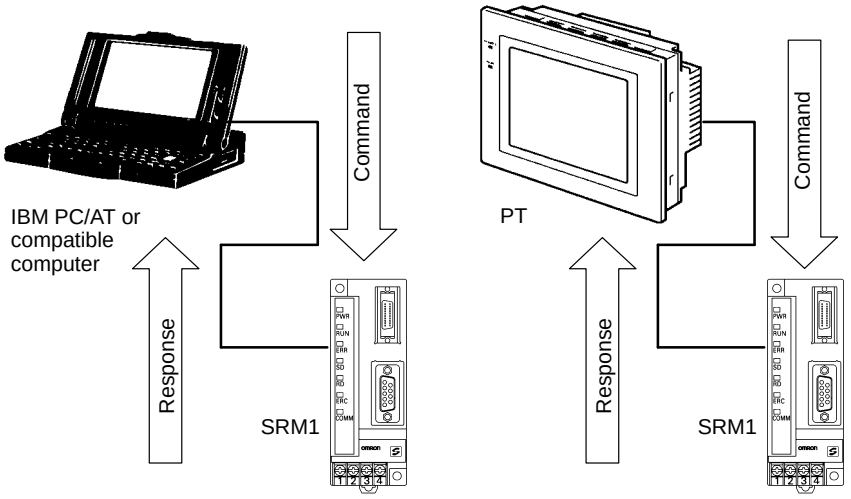
Host Link is a command/response communications system in which commands are transmitted from the host computer and corresponding responses are returned from the destination SRM1. Host Link commands can be used to read/write data in SRM1 data areas and read/write settings. Either the peripheral port or RS-232C port can be used.



Note For details on PC Setup settings, refer to *1-1 PC Setup* or *4-4-1 Host Link Communications)* in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual (W353)*.

One-to-one Host Link Connection

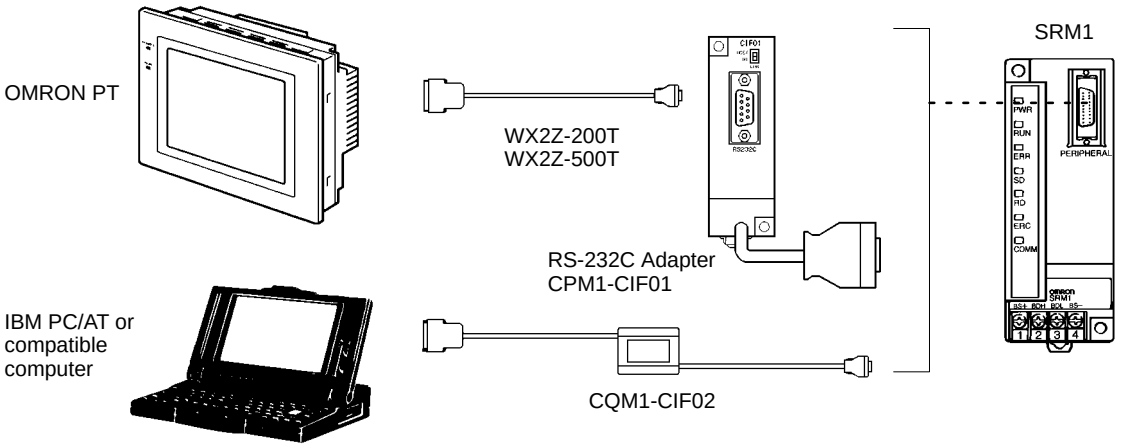
The SRM1 can be connected to an IBM PC/AT or compatible computer or a Programmable Terminal, as shown in the following diagram.



One-to-one Host Link Cables

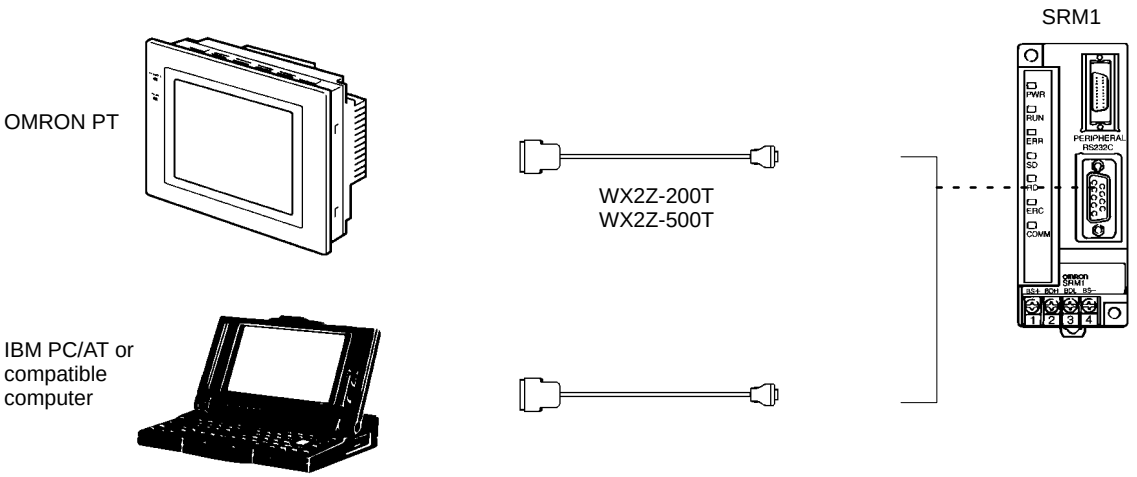
The cables differ depending on whether the peripheral port or RS-232C port is used.

Peripheral Port Connection



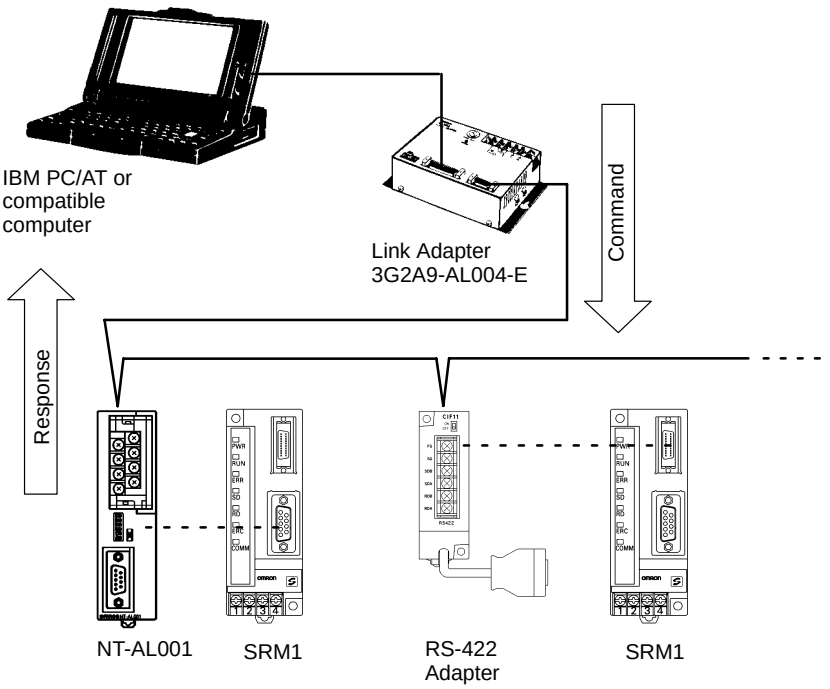
Note Set the RS-232C Adapter mode setting switch to “HOST.”

RS-232C Port Connection

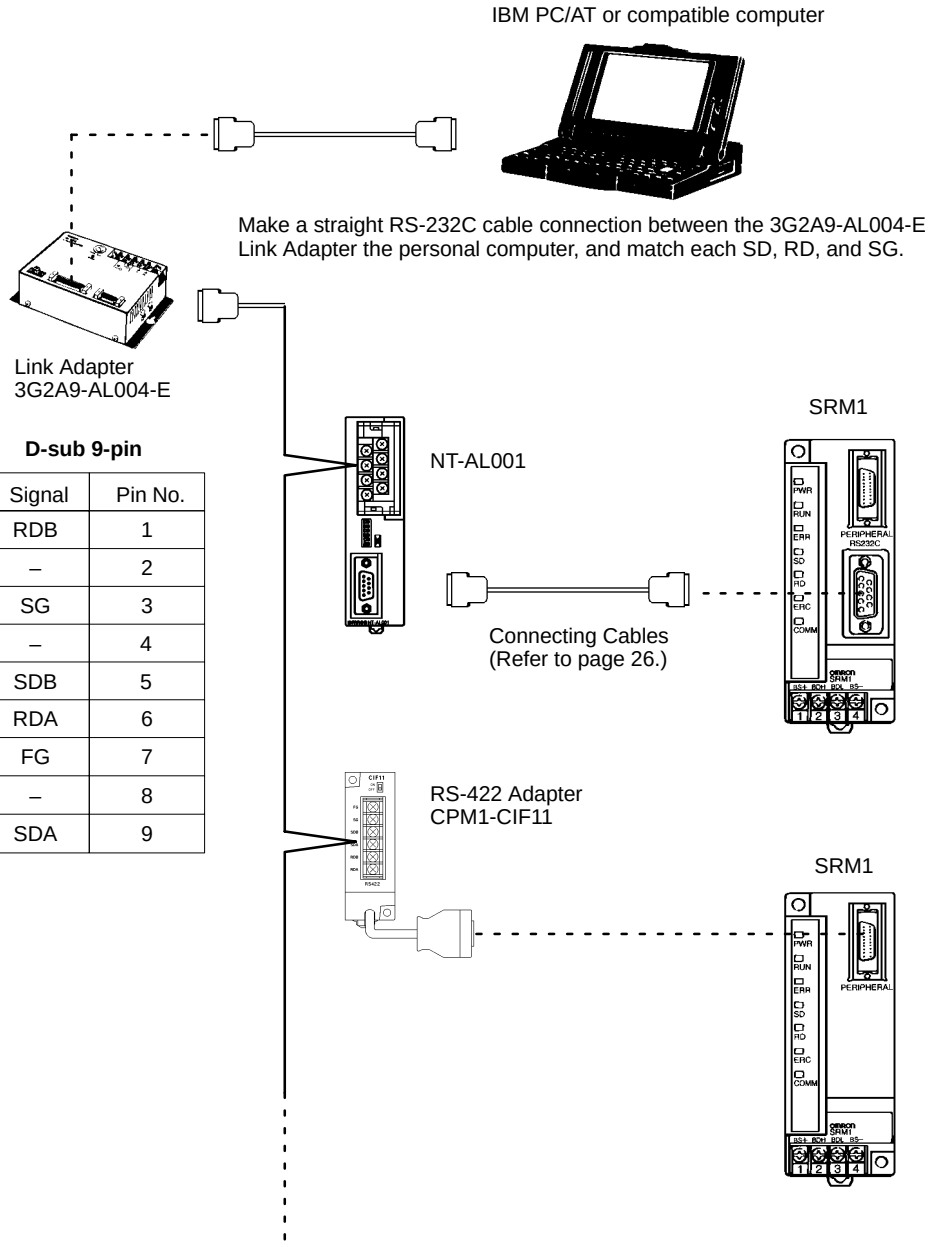


Note For details regarding RS-232C connections, refer to *3-4-4 RS-232C Port Wiring*.

One-to-N Host Link Connection



One-to-N Host Link Cables Up to 32 SRM1s can be connected to the computer via the peripheral port or RS-232C port.



D-sub 9-pin

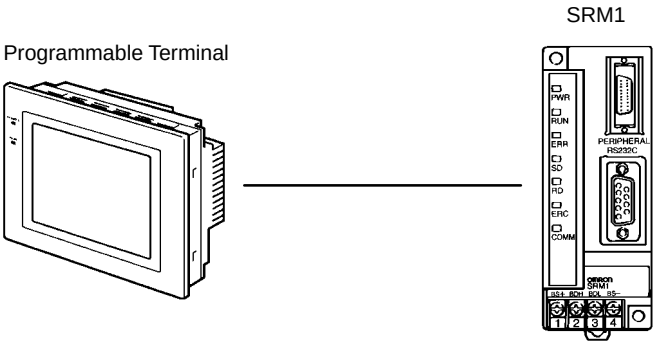
Signal	Pin No.
RDB	1
–	2
SG	3
–	4
SDB	5
RDA	6
FG	7
–	8
SDA	9

Note

1. The maximum total length of the RS-422 cable is 500 meters.
2. Turn ON the termination resistance switches for only the Link Adapters or RS-422 Adapters at both ends of the Host Link network.
3. Crimp-type terminals must be used for Link Adapter and RS-422 Adapter terminal wiring. For details, refer to *3-4-1 General Precautions for Wiring*.

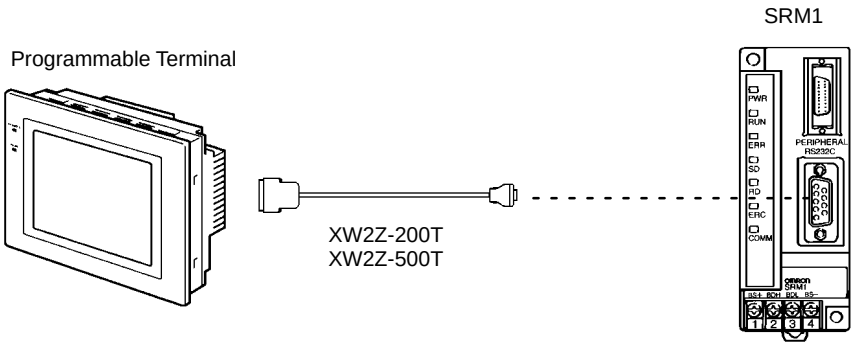
3-5 One-to-one NT Link

High-speed communications can be achieved by providing a direct access through the use of the 1:1 NT Link between the SRM1 and Programmable Terminal (PT). The RS-232C port can be used for NT Link. A 1:1 NT Link is only possible with a Master Control Unit that has an RS-232C port (SRM1-C02, SRM1-C02-V1, or SRM1-C02-V2).



NT Link Cable Connections

The SRM1 can be connected to a PT via the RS-232C port, as shown in the following illustration. For details regarding RS-232C connections, refer to *3-4-4 RS-232C Port Wiring*.

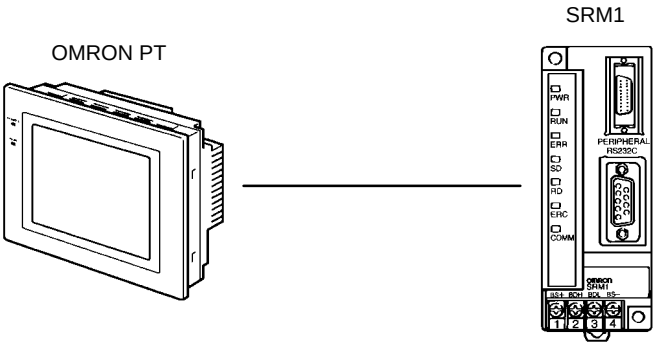


PC Setup Settings

Set the RS-232C port's communications mode to 1:1 NT Link in DM 6645 in the PC Setup. For details, refer to *1-1 PC Setup* or *4-4-3 One-to-one NT Link Communications* in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual (W353)*.

3-6 One-to-N NT Link

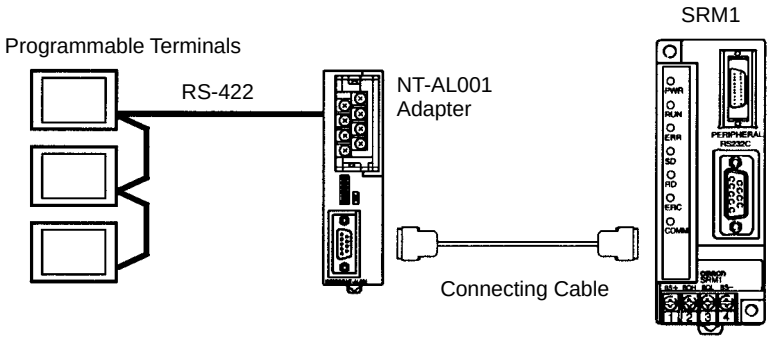
The 1:N NT Link allows an SRM1-C02-V2 PC to be connected to as many as 8 OMRON Programmable Terminals (PTs) and direct access provides high-speed communications. The RS-232C port is used to make the 1:N NT Link.



The 1:N NT Link is possible only with the SRM1-C02-V2 PCs, which have an RS-232C port.

Cable Connections

The SRM1 can be connected to OMRON PTs via the RS-232C port, as shown in the following illustration. OMRON PTs that support the 1:N NT Link must be used.



Note
1. For details on RS-422A connections, refer to the Programmable Terminal's Operation Manual.
2. For details on RS-232C connections, refer to *3-4-4 RS-232C Port Wiring*.

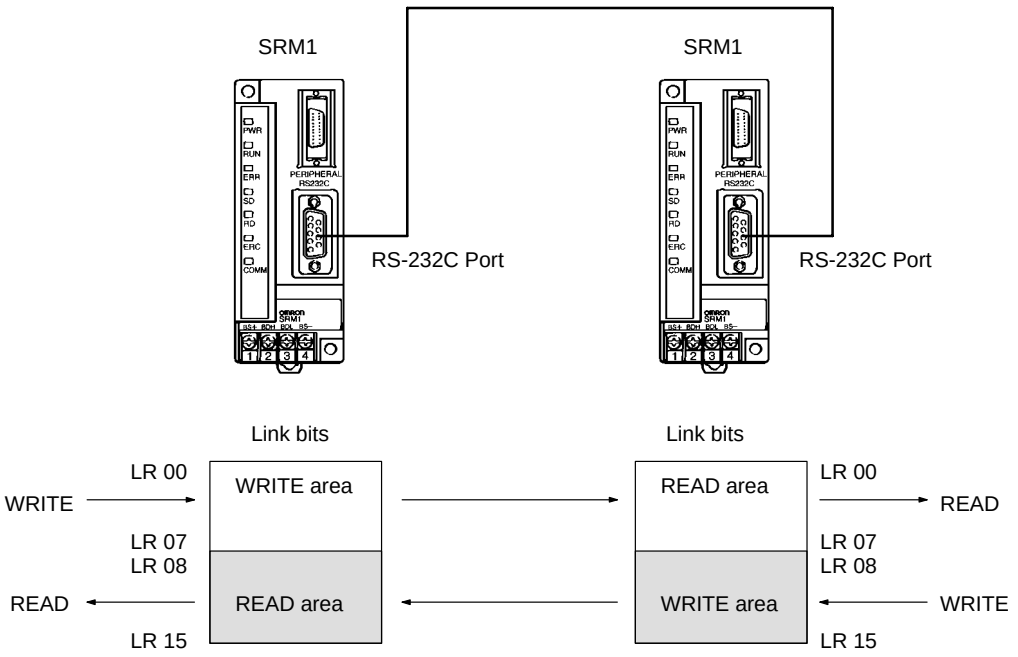
PC Setup Settings

Set the RS-232C port's communications mode to 1:N NT Link in DM 6645 in the PC Setup. For details, refer to *1-1 PC Setup* or *4-4-4 One-to-N NT Link Communications* in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual* (W353).

3-7 One-to-one PC Link Connections

3-7-1 Basics

An SRM1 can be linked one-to-one to an SRM1, CQM1, CPM1, CPM1A, CPM2A, CPM2C, C200HS, or C200HX/HE/HG PC. One PC acts as the Master and the other as the Slave to link up to 256 bits in the LR area (LR 0000 to LR 1515). In the following example, an SRM1 is linked one-to-one to another SRM1.

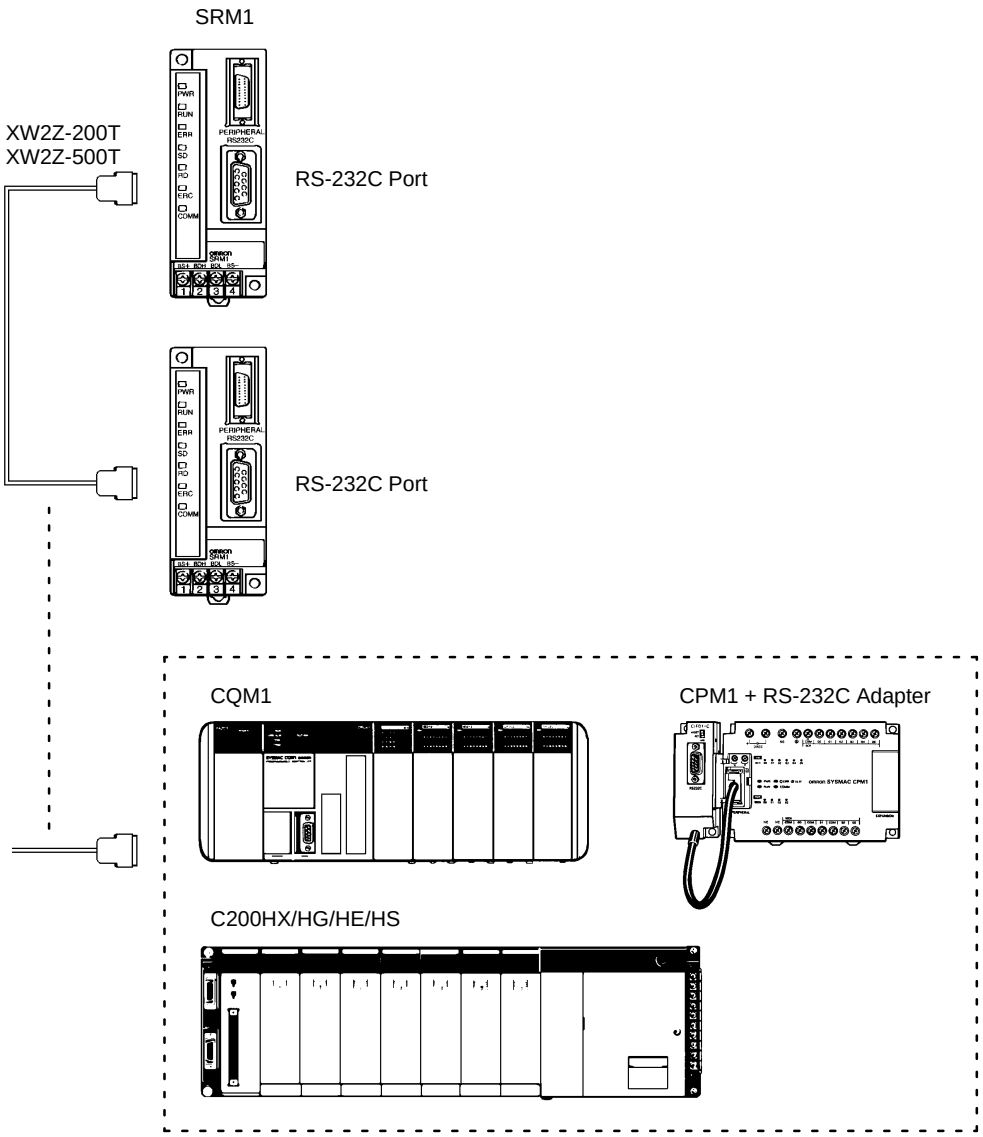


3-7-2 Restrictions

- Only the SRM1-C02-V2, which has an RS-232C port, can be used for a 1:1 PC Link.
- The only SRM1 words that can be used for link relay are the 16 words from LR 00 to LR 15. Therefore, these words must also be used at the CQM1 or C200HX/HE/HG/HS when linking any of these PCs one-to-one with an SRM1. It is not possible for words LR 16 to LR 63 to be linked one-to-one with an SRM1.

3-7-3 Cable Connections

Use RS-232C cable to connect an SRM1 with another SRM1, CQM1, CPM1, CPM1A, CPM2A, CPM2C, C200HS, or C200HX/HE/HG Programmable Controller.



Note For details regarding RS-232C connections, refer to *3-4-4 RS-232C Port Wiring*.

3-7-4 PC Setup Settings

Set the RS-232C port's communications mode to 1:1 PC Link (Slave) or 1:1 PC Link (Master) in DM 6645 in the PC Setup. For details, refer to *1-1 PC Setup* or *4-4-5 One-to-one PC Link Communications* in the *CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual (W353)*.

SECTION 4
Using the Programming Console

This section explains how to use the Programming Console. Be sure to read this section carefully if you are not already familiar with Programming Console operations.

4-1 Basic Operations 34
4-1-1 Compatible Programming Consoles 34
4-1-2 Connecting the Programming Console 35
4-1-3 Changing the SRM1's Mode 35
4-2 Programming Console Operations 36
4-2-1 Overview 36
4-2-2 Clearing Memory 37
4-2-3 Reading/Clearing Error Messages 38
4-2-4 Buzzer Operation 39
4-2-5 Reading UM Area Information 40
4-2-6 Setting Expansion Instructions 40
4-2-7 Setting and Reading a Program Memory Address and Monitoring I/O Bit Status 41
4-2-8 Instruction Search 41
4-2-9 Bit Operand Search 42
4-2-10 Inserting and Deleting Instructions 43
4-2-11 Entering or Editing Programs 44
4-2-12 Checking the Program 47
4-2-13 Bit, Digit, Word Monitor 47
4-2-14 Differentiation Monitor 49
4-2-15 Binary Monitor 50
4-2-16 Three-Word Monitor 50
4-2-17 Signed Decimal Monitor 51
4-2-18 Unsigned Decimal Monitor 51
4-2-19 Three-Word Data Modification 52
4-2-20 Changing Timer, Counter SV 52
4-2-21 Hexadecimal, BCD Data Modification 53
4-2-22 Binary Data Modification 54
4-2-23 Signed Decimal Data Modification 55
4-2-24 Unsigned Decimal Data Modification 56
4-2-25 Force Set, Reset 56
4-2-26 Clear Force Set/Reset 57
4-2-27 Hex-ASCII Display Change 57
4-2-28 Displaying the Cycle Time 58

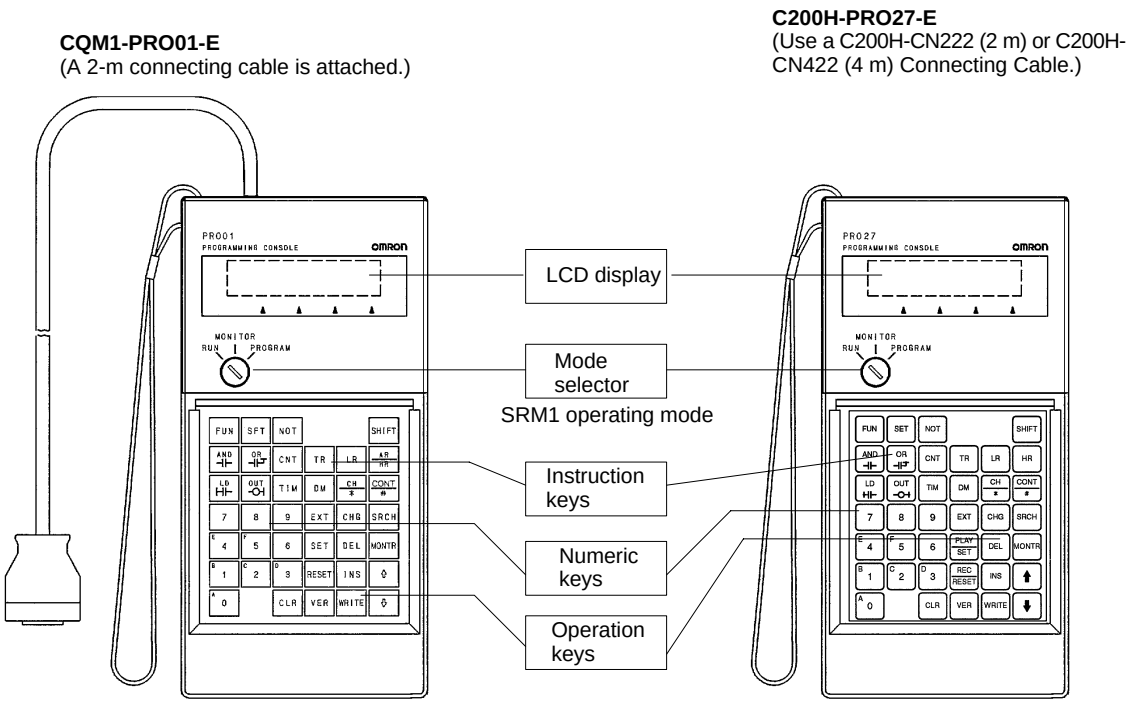
4-1 Basic Operations

This section provides information on connecting and using a Programming Console. Refer to *5-5 Programming Console Operation Errors* for details on errors that might occur during Programming Console operations.

4-1-1 Compatible Programming Consoles

There are two Programming Consoles that can be used with the SRM1: the CQM1-PRO01-E and the C200H-PRO27-E. The key functions for these Programming Consoles are identical.

Press and hold the Shift Key to input a letter shown in the upper-left corner of the key or the upper function of a key that has two functions. For example, the CQM1-PRO01-E's AR/HR key can specify either the AR or HR Area; press and release the Shift Key and then press the AR/HR Key to specify the AR Area.

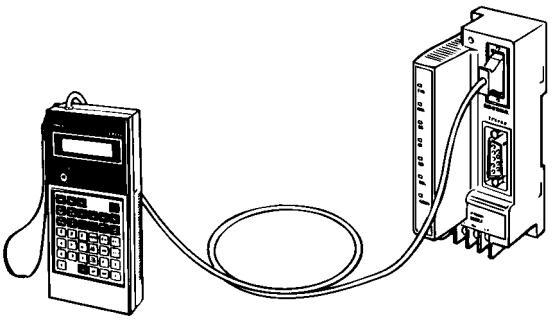


The following keys are labeled differently on the CQM1-PRO01-E and the C200H-PRO27-E, but the operation of the keys in each pair is identical.

CQM1-PRO01-E Keys	C200H-PRO27-E Keys
AR / HR	HR
SET	PLAY / SET
RESET	REC / RESET

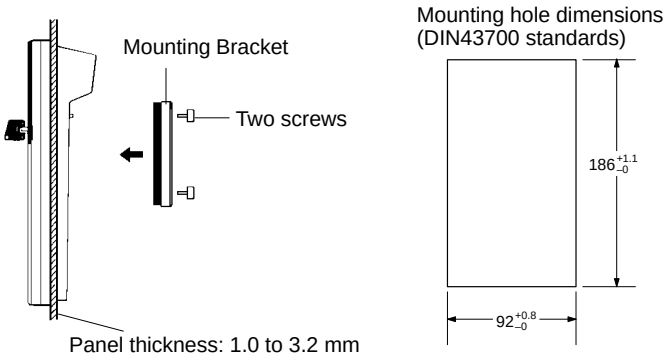
4-1-2 Connecting the Programming Console

Connect the Programming Console's connecting cable to the SRM1's peripheral port, as shown below.

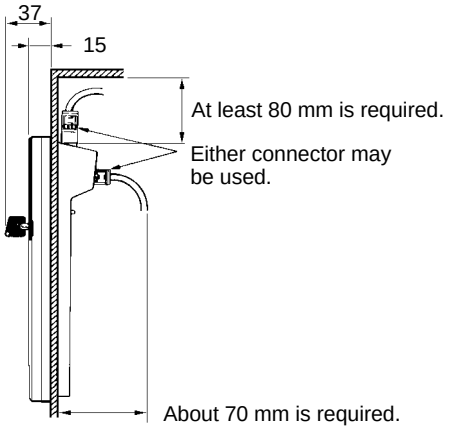


Panel Installation

The C200H-PRO27-E Programming Console can be installed in a control panel as shown in the following diagram. (The C200H-ATT01 Mounting Bracket is sold separately.)



Allow at least 80 mm for the cable connector above the Programming Console.

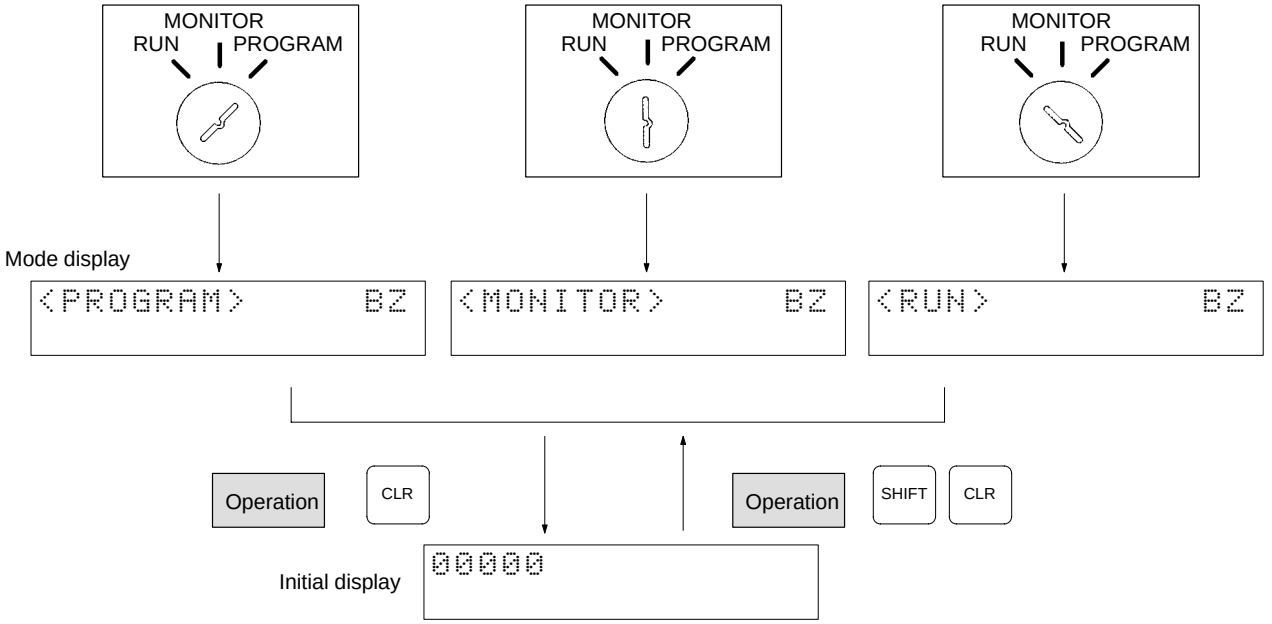


4-1-3 Changing the SRM1's Mode

Once the Programming Console has been connected, its mode switch can be used to change the SRM1's PC mode. The mode display (<PROGRAM>, <MONITOR>, or <RUN>) will appear on the Programming Console screen.

- No key operations can be performed while the mode display is displayed on the Programming Console screen. Press CLR to clear the display so that key operations can be performed.
- If the SHIFT Key is pressed while the mode switch is turned, the original display will remain on the Programming Console's screen and the mode display won't appear.

- The SRM1 will enter RUN mode automatically if a Peripheral Device such as a Programming Console isn't connected when the SRM1 is turned on (when DM 6600 is #0000).



PROGRAM Mode The SRM1 program isn't executed in PROGRAM mode. Use PROGRAM mode to create and edit the program, clear memory, or check the program for errors.

MONITOR Mode The SRM1 program is executed in MONITOR mode and I/O is processed just as it is in RUN mode. Use MONITOR mode when testing the system by monitoring the SRM1's operating status, force-setting and resetting I/O bits, changing the SV/PV of timers and counters, etc.

RUN Mode This is the SRM1's normal operating mode. The SRM1's operating status can be monitored from a Peripheral Device, but bits can't be force-set/force-reset and the SV/PV of timers and counters can't be changed.

⚠ Caution Be sure to check the system thoroughly before executing the SRM1 program to prevent any accidents that might occur when the program is first started.

4-2 Programming Console Operations

4-2-1 Overview

The following table lists the programming and monitoring operations that can be performed from a Programming Console. Refer to the rest of this section for details on operational procedures.

Name	Function
Clearing memory	Clears all or part of the Program Memory and any data areas that are not read-only, as well as the contents of the Programming Console's memory.
Reading/clearing error messages	Displays and clears error messages and displays MESSAGE instruction messages.
Buzzer operation	Turns on and off the buzzer that sounds when Programming Console keys are pressed.
Reading UM area information	Reads the state of area setting and capacity in the user memory.
Setting an expansion instruction	Reads and sets the function code assignment of an expansion instruction.
Setting a program memory address	Sets the specified program memory address when reading, writing, inserting and deleting programs.

Name	Function
Reading a program memory address	Reads the contents of the Program Memory. Displays the status of the currently displayed bit in PROGRAM and MONITOR modes.
Instruction search	Finds occurrences of the specified instruction in the program.
Bit operand search	Finds occurrences of the specified operand bit in the program.
Inserting and deleting instructions	Inserts or deletes instructions from the program.
Entering or editing programs	Overwrites the contents of the current Program Memory to either input a program for the first time or to change a program that already exists.
Checking the program	Checks for programming errors and displays the program address and error when errors are found.
Bit, digit, word monitor	Monitors the status of up to 16 bits and words, although only 3 can be shown on the display at one time.
Multiple address monitor	Monitors the status of up to 6 bits and words simultaneously.
Differentiation monitor	Monitors the up or down differentiation status of a particular bit.
Binary monitor	Monitors the ON/OFF status of any word's 16 bits.
3-word monitor	Monitors the status of three consecutive words.
Signed decimal monitor	Converts the contents of the specified word from signed hexadecimal (two's complement format) to signed decimal for display.
Unsigned decimal monitor	Converts hexadecimal data in a word to unsigned decimal for display.
3-word data modification	Changes the contents of one or more of the 3 consecutive words displayed in the 3-Word Monitor operation.
Changing timer, counter SV 1	Changes the SV of a timer or counter.
Changing timer, counter SV 2	Makes fine adjustment changes to the SV of the timer or counter.
Hexadecimal, BCD data modification	Changes the BCD or hexadecimal value of a word being monitored.
Binary data modification	Changes the status of a word's bits when the word is being monitored.
Signed decimal data modification	Changes the decimal value of a word being monitored as signed decimal data. The contents of the specified word are converted automatically to signed hexadecimal (two's complement format.)
Unsigned decimal data modification	Changes the decimal value of a word being monitored as unsigned decimal data. A change into hexadecimal data is made automatically.
Force set/reset	Forces bits ON (force set) or OFF (force reset.)
Clear force set/reset	Restores the status of all bits which have been force set of reset.
Hex-ASCII display change	Converts word data displays back and forth between 4-digit hexadecimal data and ASCII.
Displaying the cycle time	Displays the current average cycle time (scan time.)

4-2-2 Clearing Memory

This operation is used to clear all or part of the Program Memory and any data areas that are not read-only, as well as the contents of the Programming Console's memory. This operation is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

Before beginning to program for the first time or when installing a new program, clear all areas.

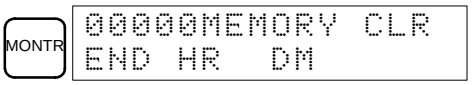
All Clear

The following procedure is used to clear memory completely.

1, 2, 3... 1. Bring up the initial display by pressing the CLR key repeatedly.

2. Press the SET, NOT, and then the RESET Key to begin the operation.

3. Press the MONTR Key to clear memory completely.



⚠ Caution The PC Setup (DM 6600 through DM 6655) will be cleared when this operation is performed.

Partial Clear

It is possible to retain the data in specified areas or part of the Program Memory. To retain the data in the HR, TC, or DM Areas, press the appropriate key after pressing SET, NOT, and RESET. Any data area that still appears on the display will be cleared when the MONTR Key is pressed.

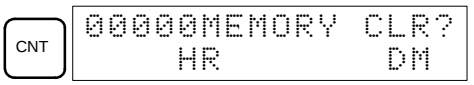
The HR Key is used to specify both the AR and HR Areas, the CNT Key is used to specify the entire timer/counter area, and the DM Key is used to specify the DM Area.

It is also possible to retain a portion of the Program Memory from the first memory address to a specified address. After designating the data areas to be retained, specify the first Program Memory address to be cleared. For example, input 030 to leave addresses 000 to 029 untouched, but to clear addresses from 030 to the end of Program Memory.

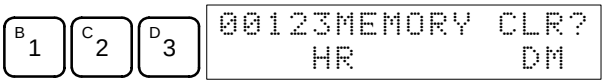
As an example, follow the procedure below to retain the timer/counter area and Program Memory addresses 000 through 122:

1, 2, 3...

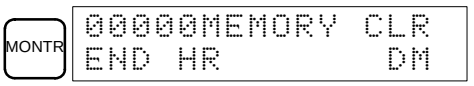
1. Press the CLR Key to bring up the initial display.
2. Press the SET, NOT, and then the RESET Key to begin the operation.
3. Press the CNT Key to remove the timer/counter area from the data areas shown on the display.



4. Press 123 to specify 123 as the starting program address.



5. Press the MONTR Key to clear the specified regions of memory.



4-2-3 Reading/Clearing Error Messages

This operation is used to display and clear error messages. It is possible to display and clear non-fatal errors and MESSAGE instruction messages in any mode, but fatal errors can be displayed and cleared in PROGRAM mode only.

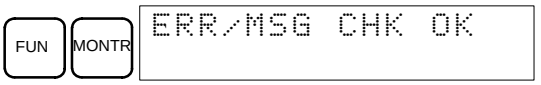
RUN	MONITOR	PROGRAM
OK	OK	OK

Before inputting a new program, any error messages recorded in memory should be cleared. It is assumed here that the causes of any of the errors for which error messages appear have already been taken care of. If the buzzer sounds when an attempt is made to clear an error message, eliminate the cause of the error, and then clear the error message. (Refer to *Section 5 Test Runs and Error Processing* for troubleshooting information.)

Key Sequence Follow the procedure below to display and clear messages.

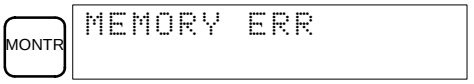
1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the FUN and then the MONTR Key to begin the operation. If there are no messages, the following display will appear:

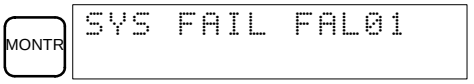


If there are messages, the most serious message will be displayed when the MONTR Key is pressed. Pressing MONTR again will clear the present message and display the next most serious error message. Continue pressing MONTR until all messages have been cleared. These are some examples of error messages:

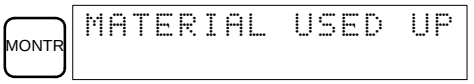
A memory error:



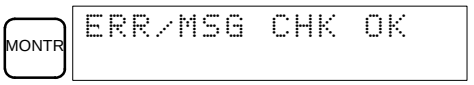
A system error:



A message:



All messages cleared:



⚠ Caution Check to be sure that no equipment is affected when turning the SRM1's power supply on or off, or when entering the password. Be careful not to cause any accidents when starting or stopping SRM1 operation.

4-2-4 Buzzer Operation

This operation is used to turn on and off the buzzer that sounds when Programming Console keys are pressed. This buzzer will also sound whenever an error occurs during PC operation. Buzzer operation for errors is not affected by this setting.

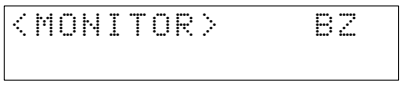
This operation is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

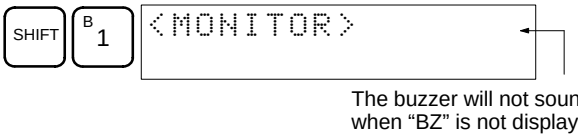
Key Sequence Follow the procedure below to turn the key-input buzzer on and off.

1, 2, 3...

1. Press the CLR, SHIFT, and then the CLR Key to bring up the mode display. In this case the PC is in PROGRAM mode and the buzzer is on.



2. Press the SHIFT and then the 1 Key to turn off the buzzer.



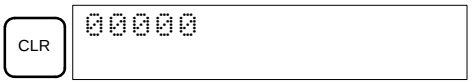
3. Press the SHIFT and then the 1 Key again to turn the buzzer back on.

4-2-5 Reading UM Area Information

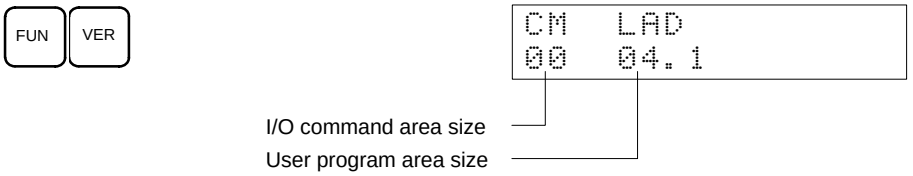
This operation is used to display the status of settings in the User Memory (UM) area, and its capacity.

RUN	MONITOR	PROGRAM
OK	OK	OK

1, 2, 3... 1. Press the CLR Key to bring up the initial display.



2. Press the FUN and VER Keys.



4-2-6 Setting Expansion Instructions

This operation is used to read and change the function codes assigned to certain instructions.

Set the expansion instructions before inputting the program. The SRM1 will not operate properly if the function codes in the program are assigned incorrectly.

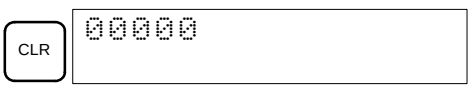
When setting the expansion instructions, set the PC Setup's DM 6602 bits 8 through 11 to "1."

Two function codes cannot be assigned to a single instruction.

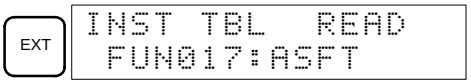
For the SRM1 expansion instruction default settings, refer to the */CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programming Manual (W353)*.

	RUN	MONITOR	PROGRAM
Read	OK	OK	OK
Set	No	No	OK

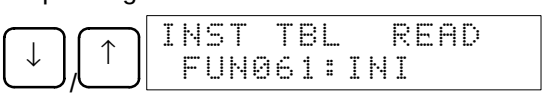
1, 2, 3... 1. Press the CLR Key to bring up the initial display.



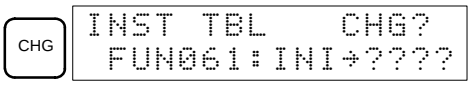
2. Press the EXT Key to display the settings.



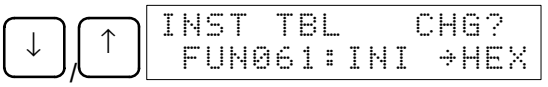
3. Use the Up and Down Arrow Keys to scroll through the function codes and read their corresponding instructions.



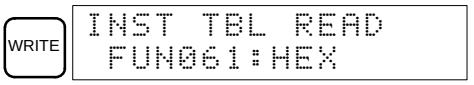
4. To change the function code assignment, press the CHG Key.



5. Use the Up and Down Arrow Keys to display the available instructions.



6. Press the WRITE Key to make the setting.



4-2-7 Setting and Reading a Program Memory Address and Monitoring I/O Bit Status

This operation is used to display the specified program memory address and is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

When a program is input for the first time, it is generally written to Program Memory starting from address 000. Because this address appears when the display is cleared, it is not necessary to specify it.

When inputting a program starting from other than 000 or to read or modify a program that already exists in memory, the desired address must be designated.

The ON/OFF status of any displayed bit will be shown if the PC is in RUN or MONITOR mode.

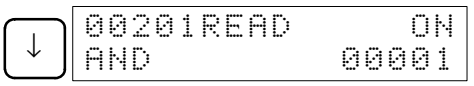
1, 2, 3... 1. Press the CLR Key to bring up the initial display.

2. Input the desired address. It is not necessary to input leading zeroes.

3. Press the Down Arrow Key.

In the RUN or MONITOR mode, ON/OFF status of the bit will be displayed.

4. Press the Up and Down Arrow Keys to scroll through the program.



4-2-8 Instruction Search

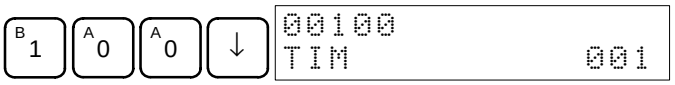
This operation is used to find occurrences of the specified instruction in the program and is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

The ON/OFF status of any displayed bit will be shown if the PC is in RUN or MONITOR mode.

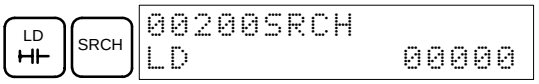
1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Input the address from which the search will begin and press the Down Arrow Key. It is not necessary to input leading zeroes.

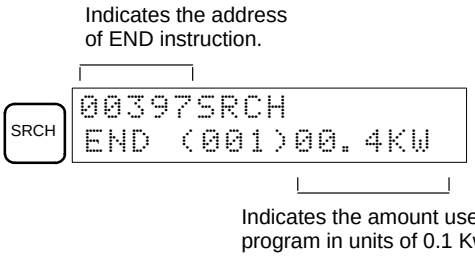


3. Input the instruction to be found and press the SRCH Key. In this case, the search is for LD instructions.

   In this case, the next LD instruction is at address 200, as shown below.



4. Press the Down Arrow Key to display the instruction's operands or press the SRCH Key to search for the next occurrence of the instruction.
5. The search will continue until an END instruction or the end of Program Memory is reached. In this case, an END instruction was reached at address 397.



4-2-9 Bit Operand Search

This operation is used to find occurrences of the specified operand bit in the program and is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

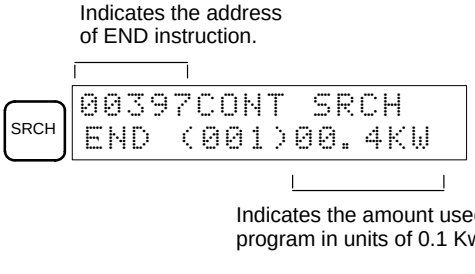
The status of any displayed bit will be shown if the PC is in RUN or MONITOR mode.

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Input the operand address. It is not necessary to input leading zeroes.

3. Press the SRCH Key to begin the search.

4. Press the SRCH Key to search for the next occurrence of the operand bit.
5. The search will continue until an END instruction or the end of Program Memory is reached. In this case, an END instruction was reached.



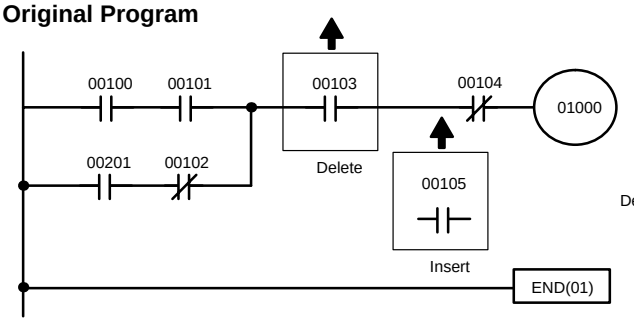
4-2-10 Inserting and Deleting Instructions

This operation is used to insert or delete instructions from the program. It is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

To demonstrate this operation, an IR 00105 NO condition will be inserted at program address 00206 and an IR 00103 NO condition deleted from address 00205, as shown in the following diagram.

Original Program



Address	Instruction	Operands
⋮	⋮	⋮
00205	AND	00103
00206	AND NOT	00104
00207	OUT	01000
00208	END(01)	-

Delete ← 00205; Insert ← 00103

Insertion

Follow the procedure below to insert the IR 00105 NO condition at address 00206.

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Input the address where the NO condition will be inserted and press the Down Arrow Key. It is not necessary to input leading zeroes.

[C 2] [A 0] [6] [↓]

```
00206READ
AND NOT     00104
```

3. Input the new instruction and press the INS Key.

[AND ⊣⊢] [B 1] [A 0] [F 5] [INS]

```
00206 INSERT?
AND         00105
```

4. Press the Down Arrow Key to insert the new instruction.

[↓]

```
00207 INSERT END
AND NOT     00104
```

Note After inserting more than one word instruction, specify the set values (operands). After inputting the set values, press the WRITE Key.

Deletion

Follow the procedure below to delete the IR 00103 NO condition at address 00205.

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Input the address where the NO condition will be deleted and press the Down Arrow Key. It is not necessary to input leading zeroes.

[C 2] [A 0] [F 5] [↓]

```
00205READ
AND         00103
```

3. Press the DEL Key.

[DEL]

```
00205 DELETE?
AND         00103
```

4. Press the Up Arrow Key to delete the specified instruction.

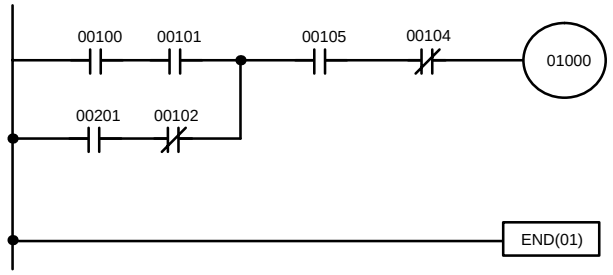
   If the instruction has more operands, the operands will be deleted automatically with the instruction.

[↑]

```
00205 DELETE END
AND         00105
```

After completing the insertion and deletion procedures, use the Up and Down Arrow Keys to scroll through the program and verify that it has been changed correctly, as shown in the following diagram.

Corrected Program



Address	Instruction	Operands
⋮	⋮	⋮
00205	AND	00105
00206	AND NOT	00104
00207	OUT	01000
00208	END(01)	-

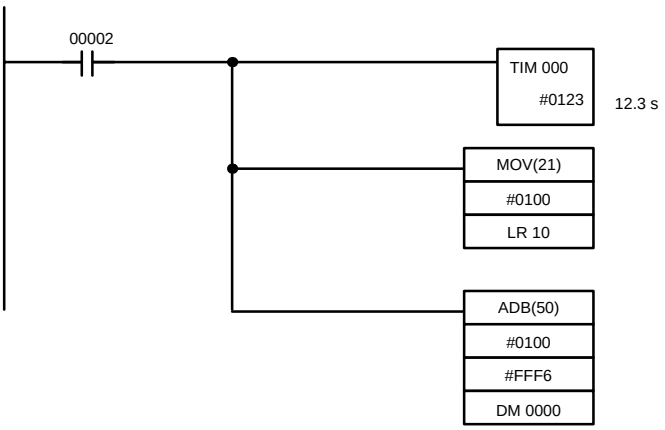
4-2-11 Entering or Editing Programs

This operation is used to enter or edit programs. It is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

The same procedure is used to either input a program for the first time or to change a program that already exists. In either case, the current contents of Program Memory is overwritten.

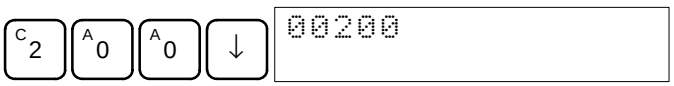
The program shown in the following diagram will be entered to demonstrate this operation.



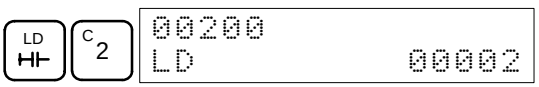
Address	Instruction	Operands
00200	LD	IR 00002
00201	TIM	000
		#0123
00202	MOV(21)	
		#0100
		LR 10
00203	ADB(50)	
		#0100
		#FFF6
		DM 0000

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Specify the address where the program will begin.
3. Input the address where the program will begin and press the Down Arrow Key. It is not necessary to input leading zeroes.



4. Input the first instruction and operand.

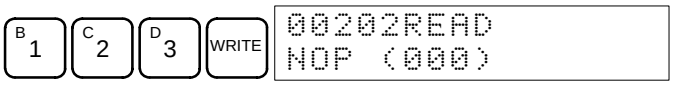


5. Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

If a mistake was made inputting the instruction, press the Up Arrow Key to return to the previous program address and input the instruction again. The mistaken instruction will be overwritten

6. Input the second instruction and operand. (In this case it isn't necessary to enter the timer number, because it's 000.) Press the WRITE Key to write the instruction to Program Memory.

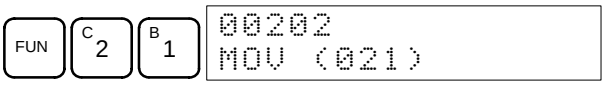
7. Input the second operand (123 to specify 12.3 seconds) and press the WRITE Key. The next program address will be displayed.



If a mistake was made inputting the operand, press the Up Arrow Key to return to display the mistaken operand, press the CONT/# Key and 123 again. The mistaken operand will be overwritten.

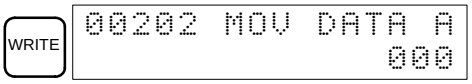
Note Counters are input in the same basic way as timers except the CNT Key is pressed instead of the TIM Key.

8. Input the third instruction and its operands. First input the instruction by pressing the FUN Key and then the function code (21 in this case).



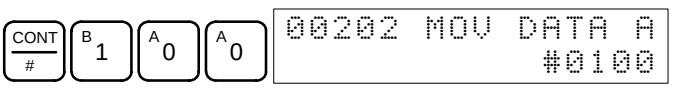
Note To input a differentiated instruction, press the NOT Key after entering the function code. The "@" symbol will be displayed next to differentiated instructions. Press the NOT Key again to change back the instruction back to a non-differentiated instruction. The "@" symbol will disappear. To change an instruction after it has been entered, simply scroll through the program until the desired instruction is displayed and press the NOT Key. The "@" symbol should be displayed next to the instruction.

9. Press the WRITE Key to write the instruction to Program Memory. The input display for the first operand will be displayed.

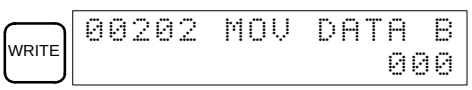


- **Writing Hexadecimal, BCD Constant**

10. Input the first operand.

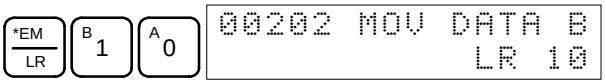


Press the WRITE Key to write the instruction to Program Memory. The input display for the second operand will appear.



• Writing a Word Address

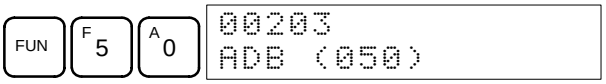
11. Input the second operand.



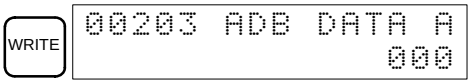
Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

Note When an instruction operand is input, the bit or word designation can be omitted.

12. Input the next instruction.

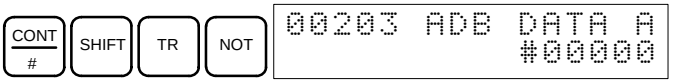


Press the WRITE Key to write the instruction to Program Memory.

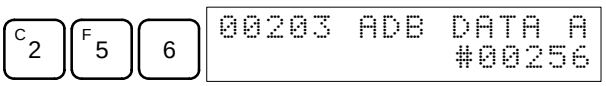


• Writing an Unsigned Decimal Number

13. The first operand is input as an unsigned integer.

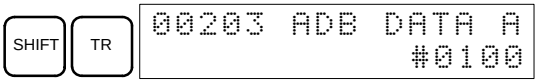


Input the value of the operand from 0 to 65535.

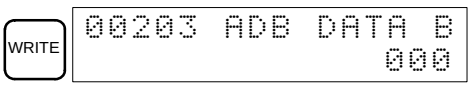


Note If an erroneous input is made, press the CLR Key to restore the status prior to the input. Then enter the correct input.

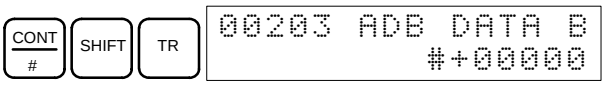
14. Restore the hexadecimal display.



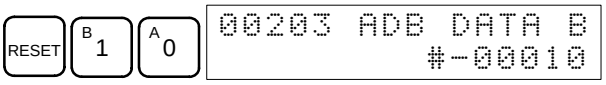
Note If an input is made outside of the permissible range, a buzzer will sound and the hexadecimal display will not be displayed.



15. The second operand is input as a signed integer.

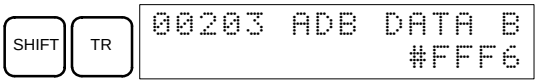


Input the value of the operand from –32,768 to 32,767. Use the SET Key to input a positive number, and use the RESET Key to input a negative number.

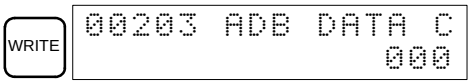


Note If an erroneous input is made, press the CLR Key to restore the status prior to the input. Then enter the correct input.

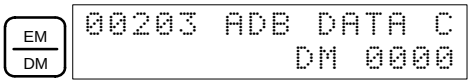
16. Restore the hexadecimal display.



Note If an input is made outside of the permissible range, a buzzer will sound and the hexadecimal display will not be displayed.



17. Input the final operand and then press the WRITE Key.



4-2-12 Checking the Program

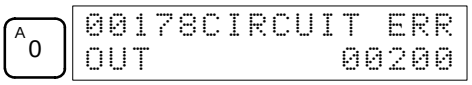
This operation checks for programming errors and displays the program address and error when errors are found. It is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the SRCH Key. An input prompt will appear requesting the desired check level.

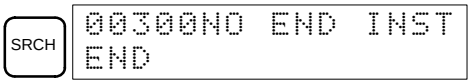
3. Input the desired check level (0, 1, or 2). The program check will begin when the check level is input, and the first error found will be displayed.



Note Refer to *5-6 Programming Errors* for details on check levels.

4. Press the SRCH Key to continue the search. The next error will be displayed. Continue pressing the SRCH Key to continue the search.

   The search will continue until an END instruction or the end of Program Memory is reached. A display like this will appear if the end of Program Memory is reached:



   A display like this will appear if an END instruction is reached:

Error Messages

For a listing of program check error messages, refer to *Section 5 Test Runs and Error Processing*.

4-2-13 Bit, Digit, Word Monitor

This operation is used to monitor the status of up to 16 bits and words, although only three can be shown on the display at any one time. Operation is possible in any mode.

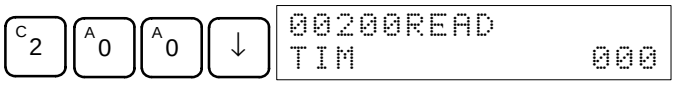
RUN	MONITOR	PROGRAM
OK	OK	OK

Program Read then Monitor

When a program address is being displayed, the status of the bit or word in that address can be monitored by pressing the MONTR Key.

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Input the desired program address and press the Down Arrow Key.

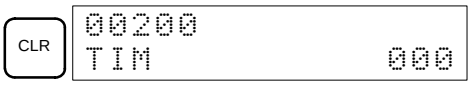


3. Press the MONTR Key to begin monitoring.

If the status of a bit is being monitored, that bit's status can be changed using the Force Set/Reset operation. Refer to page 56 for details.

If the status of a word is being monitored, that word's value can be changed using the Hexadecimal/BCD Data Modification operation. Refer to page 53 for details.

4. Press the CLR Key to end monitoring.

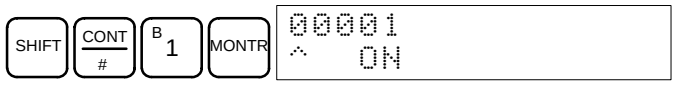


Bit Monitor

Follow the procedure below to monitor the status of a particular bit.

1, 2, 3...

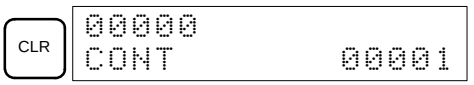
1. Press the CLR Key to bring up the initial display.
2. Input the bit address of the desired bit and press the MONTR Key.



The Up or Down Arrow Key can be pressed to display the status of the previous or next bit.

The displayed bit's status can be changed using the Force Set/Reset operation. Refer to page 56 for details.

3. Press the CLR Key to end monitoring.



Word Monitor

Follow the procedure below to monitor the status of a particular word.

1, 2, 3...

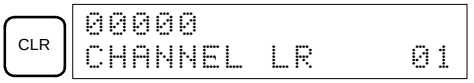
1. Press the CLR Key to bring up the initial display.
2. Input the word address of the desired word.

3. Press the MONTR Key to begin monitoring.

The Up or Down Arrow Key can be pressed to display the status of the previous or next word.

The displayed word's status can be changed using the Hexadecimal/BCD Data Modification operation. Refer to page 53 for details.

4. Press the CLR Key to end monitoring.



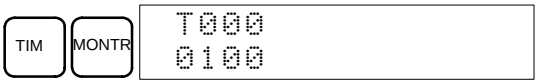
Multiple Address Monitoring

The status of up to six bits and words can be monitored simultaneously, although only three can be shown on the display at any one time.

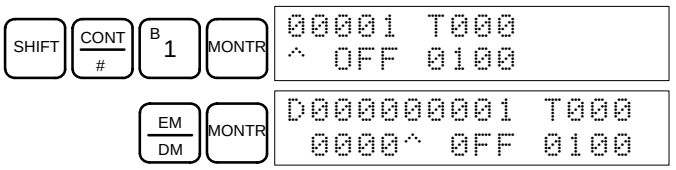
1, 2, 3...

1. Press the CLR Key to bring up the initial display.

2. Input the address of the first bit or word and press the MONTR Key.



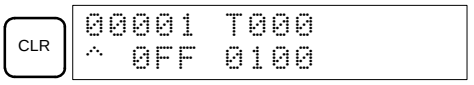
3. Repeat step 2 up to 6 times to display the next addresses to be monitored.



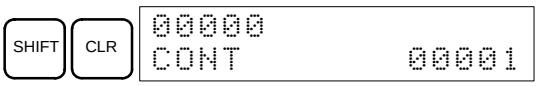
If 4 or more bits and words are being monitored, the bits and words that do not appear on the display can be displayed by pressing the MONTR Key. If the MONTR Key is pressed alone, the display will shift to the right.

If more than six bits and words are input, monitoring of the bit or word input first will be canceled.

4. Press the CLR Key to stop monitoring the leftmost bit or word and clear it from the display.



5. Press the SHIFT+CLR Keys to end monitoring altogether.



Note Press the SHIFT + CLR Keys to return to the display with the multiple address monitoring state unchanged. Press the SHIFT + MONTR Keys to display the retained multiple address monitoring state. The monitoring states can be retained for 6 bits and words.

4-2-14 Differentiation Monitor

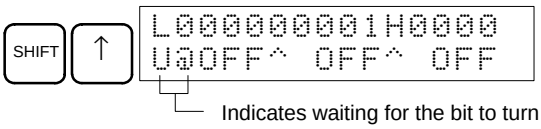
This operation is used to monitor the up or down differentiation status of a particular bit. When detected, the up or down differentiation will be displayed and the buzzer will sound. It is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

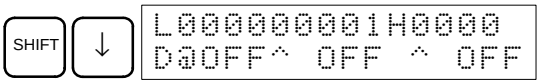
1, 2, 3... 1. Monitor the status of the desired bit according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*. If 2 or more bits are being monitored, the desired bit should be leftmost on the display.

In this case the differentiation status of LR 00 will be monitored.

2. To specify up-differentiation monitoring, press the SHIFT and then the Up Arrow Key. The symbols “U@” will appear.



To specify down-differentiation monitoring, press the SHIFT and then the Down Arrow Key. The symbols “D@” will appear.



3. The buzzer will sound when the specified bit goes from off to on (for up-differentiation) or from on to off (for down-differentiation).

```
L0000000001H0000
^ON   ^ OFF ^ OFF
```

4. Press the CLR Key to end differentiation monitoring and return to the normal monitoring display.

CLR

```
L0000000001H0000
^OFF  ^OFF ^OFF
```

4-2-15 Binary Monitor

This operation is used to monitor the ON/OFF status of any word's 16 bits. It is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

1, 2, 3...

1. Monitor the status of the desired word according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*. The desired word should be leftmost on the display if two or more words are being monitored.

```
 c010
 0000
```

(Word monitor)

2. Press the SHIFT and then the MONTR Key to begin binary monitoring. The ON/OFF status of the selected word's 16 bits will be shown along the bottom of the display. A 1 indicates a bit is on, and a 0 indicates it is off.

SHIFT MONTR

```
     c010 MONTR
0000000000000000
```

Bit 15 Bit 00

The status of force-set bits is indicated by "S," and the status of a force-reset bits is indicated by "R," as shown below.

```
    c010 MONTR
0005000000R0000
```

Force-set bit Force-reset bit

Note a) The status of displayed bits can be changed at this point. Refer to *4-2-22 Binary Data Modification* for details.

b) The Up or Down Arrow Key can be pressed to display the status of the previous or next word's bits.

3. Press the CLR Key to end binary monitoring and return to the normal monitoring display.

CLR

```
 c010
 0000
```

4-2-16 Three-Word Monitor

This operation is used to simultaneously monitor the status of three consecutive words. In RUN mode or MONITOR mode, changes are displayed as they occur. The 3-word monitor operation can be used during I/O monitoring, including multiple address monitoring. Refer to *4-2-13 Bit, Digit, Word Monitor*.

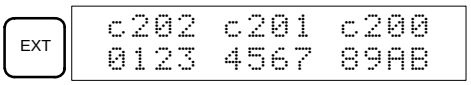
RUN	MONITOR	PROGRAM
OK	OK	OK

1, 2, 3... 1. Monitor the status of the first of the three words according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*.

If two or more words are being monitored, the desired first word should be leftmost on the display.

(Word monitor)

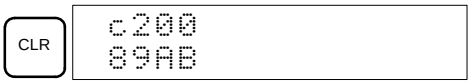
2. Press the EXT Key to begin 3-word monitoring. The status of the selected word and the next two words will be displayed, as shown below. In this case, DM 0000 was selected.



The Up and Down Arrow Keys can be used to shift one address up or down.

The status of the displayed words can be changed at this point. Refer to *4-2-19 3-word Data Modification*.

3. Press the CLR Key to end 3-word monitoring and return to the normal monitoring display. The rightmost word on the 3-word monitor display will be monitored.

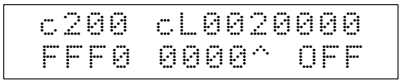


4-2-17 Signed Decimal Monitor

This operation converts the contents of the specified word from signed hexadecimal (two's complement format) to signed decimal for display. The operation can be executed while using I/O monitoring, multiple address monitoring or 3-word monitoring.

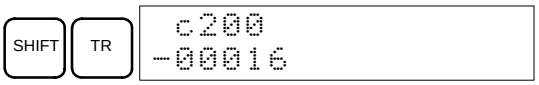
RUN	MONITOR	PROGRAM
OK	OK	OK

1, 2, 3... 1. Monitor the word that is to be used for decimal monitor with sign. During multiple address monitoring, the leftmost word will be converted.



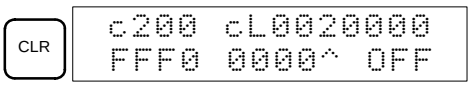
(Multiple address monitor)

2. Press the SHIFT+TR Keys to display the leftmost word as signed decimal.



At this point, the contents of the displayed word can be changed with a signed-decimal input. Refer to *4-2-23 Signed Decimal Data Modification*.

3. Press the CLR Key or the SHIFT+TR Keys to end the unsigned decimal display and return to normal monitoring.



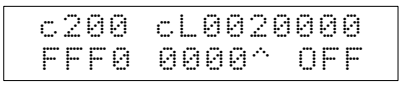
4-2-18 Unsigned Decimal Monitor

This operation is used to convert hexadecimal data in a word to unsigned decimal for display. The operation can be executed while using I/O monitoring, multiple address monitoring or 3-word monitoring.

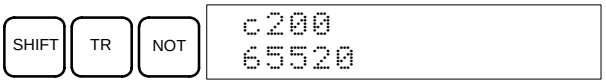
RUN	MONITOR	PROGRAM
OK	OK	OK

1, 2, 3... 1. Monitor the word that is to be used for decimal monitor without sign. During multiple address monitoring and 3-word monitoring, the leftmost word will be converted.

Multiple address monitoring

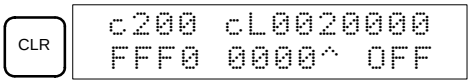


2. Press the SHIFT+TR+NOT Keys to display the leftmost word as unsigned decimal.



At this point, the contents of the displayed word can be changed with an unsigned-decimal input. Refer to *4-2-24 Unsigned Decimal Data Modification*.

3. Press the CLR Key or the SHIFT+TR Keys to end the unsigned decimal display and return to normal monitoring.



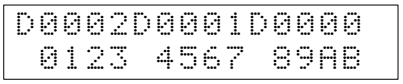
4-2-19 Three-Word Data Modification

This operation is used to change the contents of one or more of the 3 consecutive words displayed in the 3-Word Monitor operation. It is possible in MONITOR or PROGRAM mode only.

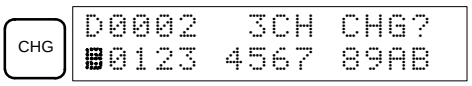
RUN	MONITOR	PROGRAM
No	OK	OK

1, 2, 3... 1. Monitor the status of the desired words according to the procedure described *4-2-16 3-Word Monitor*.

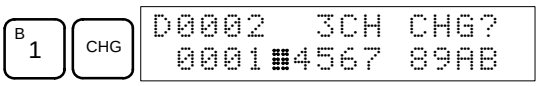
(3-word monitor)



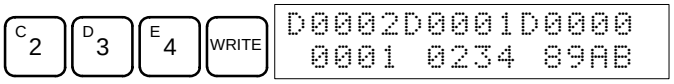
2. Press the CHG Key to begin 3-word data modification. The cursor will appear next to the contents of the leftmost word.



3. Input the new value for the leftmost word on the display and press the CHG Key if more changes will be made.
(Input the new value and press the WRITE Key to write the changes in memory if no more changes will be made.)



4. Input the new value for the middle word on the display and press the CHG Key if the rightmost word will be changed. Input the new value and press the WRITE Key to write the changes in memory if the rightmost word will not be changed. (In this case, it will not.)



Note If the CLR Key is pressed before the WRITE Key, the operation will be cancelled and the 3-word monitor display will return without any changes in data memory.

4-2-20 Changing Timer, Counter SV

There are two operations that can be used to change the SV of a timer or counter. They are possible in MONITOR or PROGRAM mode only. In MONITOR mode, the SV can be changed while the program is being executed.

RUN	MONITOR	PROGRAM
No	OK	OK

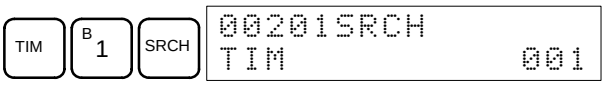
The timer or counter SV can be changed either by inputting a new value or by incrementing or decrementing the current SV.

Inputting a New SV Constant

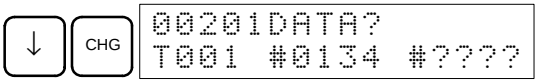
This operation can be used to input a new SV constant, as well as to change an SV from a constant to a word address designation and vice versa. The following examples show how to input a new SV constant and how to change the SV from a constant to an address.

1, 2, 3...

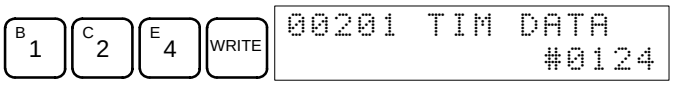
1. Press the CLR Key to bring up the initial display.
2. Display the desired timer or counter.



3. Press the Down Arrow Key and then the CHG Key.



4. At this point a new SV constant can be input or the SV constant can be changed to a word address designation
   a) To input a new SV constant, input the constant and press the WRITE Key.



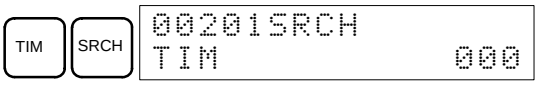
   b) To change to a word address designation, input the word address and press the WRITE Key.

Incrementing and Decrementing a Constant

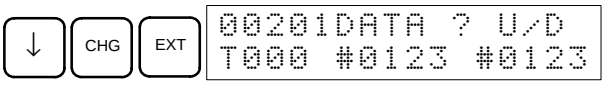
This operation can be used to increment and decrement an SV constant. It is possible only when the SV has been entered as a constant.

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Display the desired timer or counter.



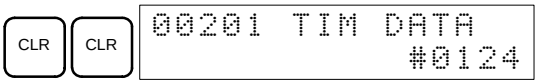
3. Press the Down Arrow, CHG, and then the EXT Key.



   The constant on the left is the old SV and the constant on the right will become the new SV constant in step 5.

4. Press the Up and Down Arrow Keys to decrement and increment the constant on the right. (In this case the SV is incremented once.)

5. Press the CLR Key twice to change the timer's SV to the new value.



4-2-21 Hexadecimal, BCD Data Modification

This operation is used to change the BCD or hexadecimal value of a word being monitored using the procedure described in *4-2-13 Bit, Digit, Word Monitor*. It is possible in MONITOR or PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	OK	OK

Words SR 253 to SR 255 cannot be changed.

⚠ Caution Before changing the contents of I/O memory, be sure that the changes will not cause equipment to operate unexpectedly or dangerously. In particular, take care when changing the status of output bits. The PC continues to refresh I/O bits even if the PC is in PROGRAM mode, so devices connected to output points may operate unexpectedly.

1, 2, 3...

1. Monitor the status of the desired word according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*. If two or more words are being monitored, the desired word should be leftmost on the display.

(Word monitor)

```
D0000
 0119
```

2. Press the CHG Key to begin hexadecimal, BCD data modification.

CHG

```
     PRES VAL?
D0000 0119  ????
```

3. Input the new PV and press the WRITE Key to change the PV.

   The operation will end and the normal monitoring display will return when the WRITE Key is pressed.

C 2 A 0 A 0 WRITE

```
D0000
 0200
```

4-2-22 Binary Data Modification

This operation is used to change the status of a word's bits when the word is monitored using the procedure described in *4-2-15 Word Monitor*. It is possible in MONITOR or PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	OK	OK

Bits SR 25300 to SR 25507 and timer/counter flags cannot be changed.

⚠ Caution Before changing the contents of I/O memory, be sure that the changes will not cause equipment to operate unexpectedly or dangerously. In particular, take care when changing the status of output bits. The PC continues to refresh I/O bits even if the PC is in PROGRAM mode, so devices connected to output points may operate unexpectedly.

1, 2, 3...

1. Monitor the status of the desired word according to the procedure described *4-2-15 Word Monitor*.

(BInary monitor)

```
     c010 MONTR
1000010101010101
```

Bit 15 Bit 00

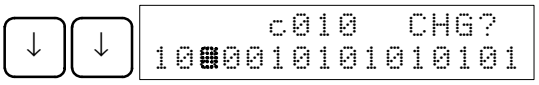
2. Press the CHG Key to begin binary data modification.

CHG

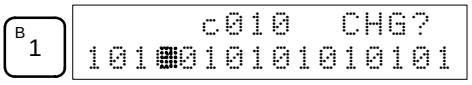
```
     c010  CHG?
▒0000101010101 01
```

A flashing cursor will appear over bit 15. The cursor indicates which bit can be changed.

3. Three sets of keys are used to move the cursor and change bit status:
   a) Use the Up and Down Arrow Keys to move the cursor to the left and right.



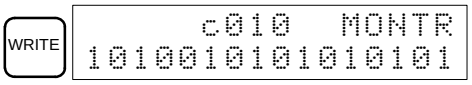
   b) Use the 1 and 0 keys to change a bit's status to on or off. The cursor will move one bit to the right after one of these keys is pressed.



   c) Use the SHIFT+SET and SHIFT+RESET Keys to force-set or force-reset a bit's status. The cursor will move one bit to the right after one of these keys is pressed. The NOT Key will clear force-set or force-reset status.

   Note Bits in the DM Area cannot be force-set or force-reset.

4. Press the WRITE Key to write the changes in memory and return to the binary monitor.



4-2-23 Signed Decimal Data Modification

This operation is used to change the decimal value of a word being monitored as signed decimal data. The contents of the specified word are converted automatically to signed hexadecimal (two's complement format).

Words SR 253 to SR 255 cannot be changed.

RUN	MONITOR	PROGRAM
No	OK	OK

⚠ Caution Before changing the contents of I/O memory, be sure that the changes will not cause equipment to operate unexpectedly or dangerously. In particular, take care when changing the status of output bits. The PC continues to refresh I/O bits even if the PC is in PROGRAM mode, so devices connected to output points may operate unexpectedly.

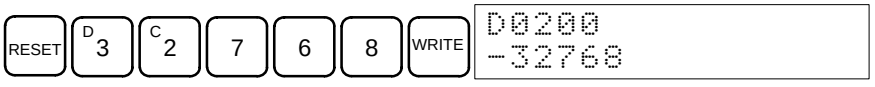
1, 2, 3... 1. Monitor (signed decimal) the status of the word for which the present value is to be changed.

(Signed decimal monitor)

2. Press the CHG Key to begin decimal data modification.

3. Input the new PV and press the WRITE Key to change the PV. The operation will end and the signed-decimal monitoring display will return when the WRITE Key is pressed.

   The PV can be set within a range of –32,768 and 32,767. Use the SET Key to input a positive number, and use the RESET Key to input a negative number.



   Press the CLR Key or the SHIFT and TR Keys to return to the normal monitoring display.

   If an erroneous input has been made, press the CLR Key to restore the status prior to the input. Then enter the correct input.

4-2-24 Unsigned Decimal Data Modification

This operation is used to change the decimal value of a word being monitored as unsigned decimal data. A change into hexadecimal data is made automatically.

Words SR 253 to SR 255 cannot be changed.

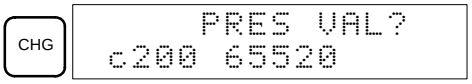
RUN	MONITOR	PROGRAM
No	OK	OK

⚠ Caution Before changing the contents of I/O memory, be sure that the changes will not cause equipment to operate unexpectedly or dangerously. In particular, take care when changing the status of output bits. The PC continues to refresh I/O bits even if the PC is in PROGRAM mode, so devices connected to output points may operate unexpectedly.

1, 2, 3... 1. Monitor (unsigned decimal) the status of the word for which the present value is to be changed.

(Unsigned decimal monitor)

2. Press the CHG Key to begin decimal data modification.



3. Input the new PV and press the WRITE Key to change the PV. The operation will end and the decimal-without-sign monitoring display will return when the WRITE Key is pressed.

   The PV can be set within a range of 0 to 65,535.

   Press the CLR Key or the SHIFT and TR Keys to return to the normal monitoring display.

   If an erroneous input has been made, press the CLR Key to restore the status prior to the input. Then enter the correct input.

4-2-25 Force Set, Reset

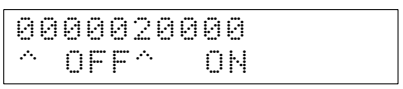
This operation is used to force bits ON (force set) or OFF (force reset) and is useful when debugging the program or checking output wiring. It is possible in MONITOR or PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	OK	OK

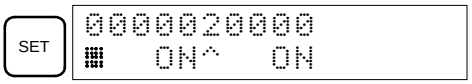
⚠ Caution Before changing the contents of I/O memory, be sure that the changes will not cause equipment to operate unexpectedly or dangerously. In particular, take care when changing the status of output bits. The PC continues to refresh I/O bits even if the PC is in PROGRAM mode, so devices connected to output points may operate unexpectedly.

1, 2, 3... 1. Monitor the status of the desired bit according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*. If two or more words are being monitored, the desired bit should be leftmost on the display.

(Multiple address monitor)



2. Press the SET Key to force the bit ON or press the RESET Key to force the bit OFF.



The cursor in the lower left corner of the display indicates that the force set/reset is in progress. Bit status will remain ON or OFF only as long as the key is held down; the original status will return one cycle after the key is released.

3. Press the SHIFT+SET or SHIFT+RESET Keys to maintain the status of the bit after the key is released. In this case, the force-set status is indicated by an "S" and the force-reset status is indicated by an "R."

To return the bit to its original status, press the NOT Key or perform the Clear Force Set/Reset operation. Refer to *4-2-26 Clear Force Set/Reset* for details.

Forced status will also be cleared when the PC's operating mode is changed (unless SR 25211 is ON, in which case forced status will not be cleared when changing from PROGRAM to MONITOR mode) or when operation stops as the result of a fatal error or power interruption.

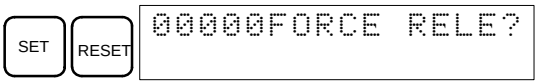
4-2-26 Clear Force Set/Reset

This operation is used to restore the status of all bits which have been force set or reset. It is possible in MONITOR or PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	OK	OK

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the SET and then the RESET Key. A confirmation message will appear.



Note If the wrong key is mistakenly pressed, press CLR and start again from the beginning.

3. Press the NOT Key to clear the force-set/reset status of bits in all data areas.

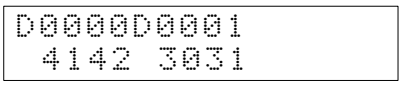
4-2-27 Hex-ASCII Display Change

This operation is used to convert word data displays back and forth between 4-digit hexadecimal data and ASCII. It is possible in any mode.

RUN	MONITOR	PROGRAM
OK	OK	OK

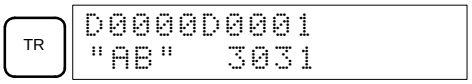
1, 2, 3...

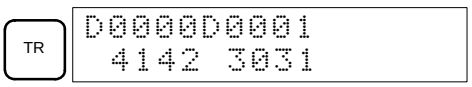
1. Monitor the status of the desired word(s) according to the procedure described in *4-2-13 Bit, Digit, Word Monitor*.



(Multiple address monitor)

2. Press the TR Key to switch to ASCII display. The display will toggle between hexadecimal and ASCII displays each time the TR Key is pressed.





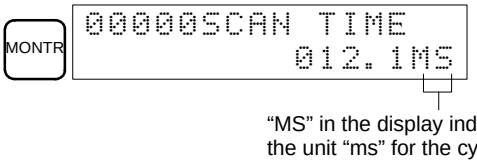
4-2-28 Displaying the Cycle Time

This operation is used to display the current average cycle time (scan time). It is possible only in RUN or MONITOR mode while the program is being executed.

RUN	MONITOR	PROGRAM
OK	OK	No

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the MONTR Key to display the cycle time.



There might be differences in displayed values when the MONTR Key is pressed repeatedly. These differences are caused by changing execution conditions.

SECTION 5
Test Runs and Error Processing

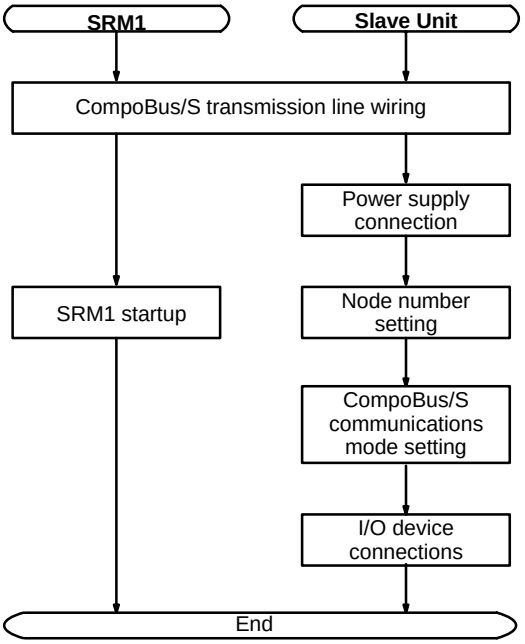
This section describes procedures for test runs of SRM1 operation, self-diagnosis functions, and error processing to identify and correct the hardware and software errors that can occur during operation.

5-1 Startup Procedure 60
5-1-1 Flowchart for Configuring and Checking the System 60
5-1-2 SRM1 Test Run Procedure 60
5-1-3 Flash Memory Precautions 61
5-2 Entering the Program 62
5-2-1 Before Programming 62
5-2-2 Clearing Memory 62
5-2-3 Ladder Programming Example 64
5-2-4 Writing the Program 68
5-2-5 Checking the Program 71
5-3 Test Run 71
5-4 Error Processing 72
5-4-1 Non-fatal Errors 72
5-4-2 Fatal Errors 72
5-4-3 Identifying Errors 73
5-4-4 User-defined Errors 73
5-4-5 Reading/Clearing Error Messages 74
5-5 Programming Console Operation Errors 75
5-6 Programming Errors 75
5-7 Troubleshooting Flowcharts 77

5-1 Startup Procedure

5-1-1 Flowchart for Configuring and Checking the System

Check the following items when configuring the system.



No.	Items to check	Contents	Reference
1	CompoBus/S transmission line wiring	Is the CompoBus/S transmission cable securely connected, and with no loose terminal screws? Is the end of the transmission path (i.e., the farthest from the SRM1) connected to a termination resistance?	Pages 23, 24 Also refer to the *CompoBus/S Operation Manual (W266)*
2	Power supply connection	Is the wiring correct? Are there any loose terminal screws? Are there any short-circuited connectors?	Pages 23, 24
3	Slave settings	Is the bit allocation correct? Have the node numbers been set properly, with no duplication?	Page 8
4	Slave I/O connections	Is the wiring correct? Are there any loose terminal screws? Are there any short-circuited connectors?	Refer to the *CompoBus/S Operation Manual (W266)*
5	CompoBus/S communications mode setting	Is the CompoBus/S communications mode setting correct?	

5-1-2 SRM1 Test Run Procedure

1, 2, 3... 1. Power Supply Application

a) Check the SRM1's power supply voltage and terminal connections.

b) Check the CompoBus/S transmission line terminal connections and the Slave's node number and power supply.

c) Check the I/O devices' power supply voltage and terminal connections.

d) Turn on the power supply. Turn on the power supply starting from the Slave.

e) Check that the "PWR" indicator lights.

f) Use the Programming Console to set the SRM1 to PROGRAM mode.

g) Use the Programming Console to set the maximum number of Slaves (bits 00 to 03 of DM 6603) and the CompoBus/S communications mode (bits 04 to 07 of DM 6603). (The SRM1's power must be turned OFF and then ON again to enable new settings in DM 6603.)

h) Check that the "SD" and "RD" indicators light.

2. I/O Wiring Checks

a) With the SRM1 in PROGRAM mode, check the output wiring by turning on the output bits with the force set and force reset operations. Refer to *4-2-25 Force Set, Reset* for details.

b) Check the input wiring with the SRM1's input indicators or the Programming Console's monitor operations. Check the input wiring with the Slaves input indicators, the Programming Console's I/O monitoring, or the Programming Console's multiple address monitoring.

3. Test Run

Use a Programming Console to set the SRM1 to RUN or MONITOR mode and check that the "RUN" indicator lights.

4. Program Input

a) Use the Programming Console, SYSMAC-CPT, or SYSMAC Support Software (SSS) to write the program.

b) Check the sequence of operation.

Note When using the SYSMAC-CPT Support Software, set the PC model to "SRM1." When using the SYSMAC Support Software, set the PC model to "CQM1."

5. Debugging

Correct any programming errors that are detected.

5-1-3 Flash Memory Precautions

The SRM1 uses the flash memory to hold the contents of the user program memory, the read-only DM area (DM 6144 through DM 6599), and PC Setup (DM 6600 through DM 6655). Observe the following precautions to protect the flash memory and ensure proper operation.

1, 2, 3...

1. Perform either one of the following to write the above contents onto the flash memory.

a) Switch the SRM1 to RUN or MONITOR mode.

b) Restart the power supply to the SRM1.

Note If the power is turned off without changing the mode after making changes in the above areas using a Peripheral Device, the changes will not be written to flash memory. Although the data in these areas is backed up by a lithium battery, the changes will be lost if the battery fails or is removed because the program in the flash memory will be automatically read into the user program memory. The standard service life of the lithium battery is ten years minimum under an ambient temperature of 25°C.

2. When the SRM1 is operated for the first time after changing the program memory, the read-only DM area (DM 6144 through DM 6599), or PC Setup (DM 6600 through DM 6655), it will take about 850 ms longer than usual before the SRM1 starts. Be sure to take this one-time startup delay into account.

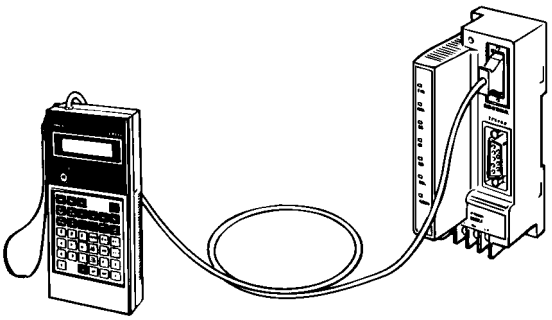
3. If one of the following three operations is performed in MONITOR or RUN mode, a “cycle time over” warning will not be issued. When performing online edit operation, take the I/O response time of the SRM1 into account. The SRM1 will extend the cycle time for up to 850 ms and interrupts will be disabled while the program or PC Setup is being overwritten.
   - Program changes with the online edit operation
   - Changes to the read-only DM area (DM 6144 through DM 6599)
   - Changes to the PC Setup (DM 6600 through DM 6655)

5-2 Entering the Program

5-2-1 Before Programming

Connecting the Programming Console

Connect the cable from the Programming Console (CQM1-PRO01-E or C200H-PRO27-E) to the SRM1’s Peripheral Port.



5-2-2 Clearing Memory

This operation is used to clear all or part of the Program Memory and any data areas that are not read-only, as well as the contents of the Programming Console’s memory. This operation is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

Before beginning to program for the first time or when installing a new program, clear all areas.

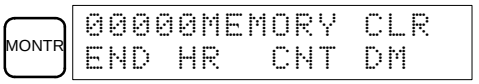
All Clear

The following procedure is used to clear memory completely.

1, 2, 3...

1. Bring up the initial display by pressing the CLR key repeatedly.
2. Press the SET, NOT, and then the RESET Key to begin the operation.

3. Press the MONTR Key to clear memory completely.



⚠ Caution Check to be sure that no equipment is affected when turning the SRM1’s power supply on or off, or when entering the password. Be careful not to cause any accidents when starting or stopping SRM1 operation.

Partial Clear

It is possible to retain the data in specified areas or part of the Program Memory. To retain the data in the HR, TC, or DM Areas, press the appropriate key after pressing SET, NOT, and RESET. Any data area that still appears on the display will be cleared when the MONTR Key is pressed.

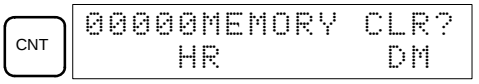
The HR Key is used to specify both the AR and HR Areas, the CNT Key is used to specify the entire timer/counter area, and the DM Key is used to specify the DM Area.

It is also possible to retain a portion of the Program Memory from the first memory address to a specified address. After designating the data areas to be retained, specify the first Program Memory address to be cleared. For example, input 030 to leave addresses 000 to 029 untouched, but to clear addresses from 030 to the end of Program Memory.

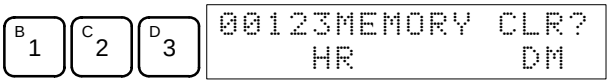
As an example, follow the procedure below to retain the timer/counter area and Program Memory addresses 000 through 122:

1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the SET, NOT, and then the RESET Key to begin the operation.
3. Press the CNT Key to remove the timer/counter area from the data areas shown on the display.



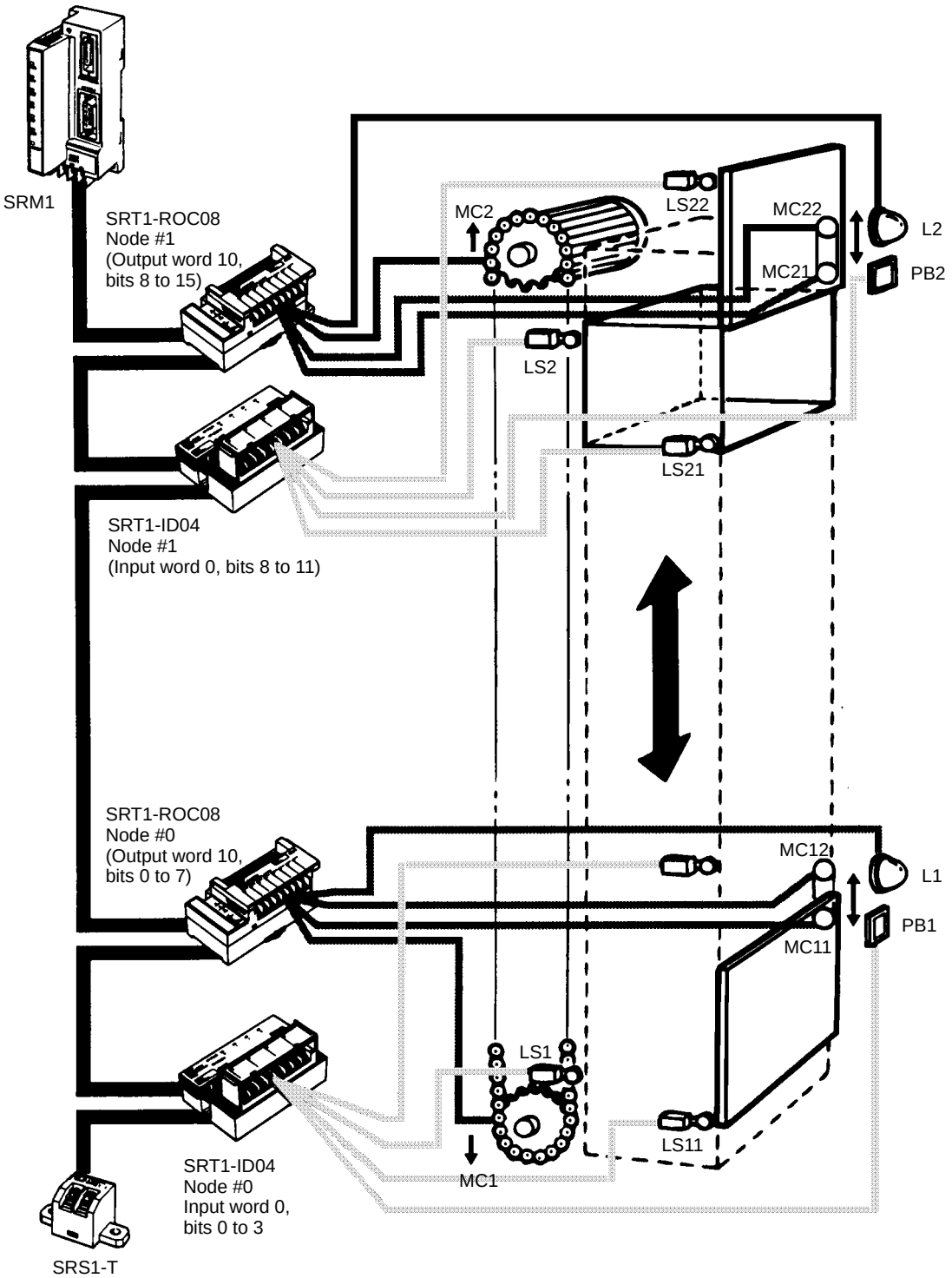
4. Press 123 to specify 123 as the starting program address.



5. Press the MONTR Key to clear the specified regions of memory.

5-2-3 Ladder Programming Example

In this example, the SRM1's distributed I/O is used to control a lift (e.g., a dumb waiter) connecting a kitchen on the first floor and a customer serving area on the second floor.



Explanation of Operations

Initially, the lift is on the first floor (where the kitchen is located) and LS1 is ON. The door is open and LS12 is ON. L1 is ON, and the first-floor indicator light is on.

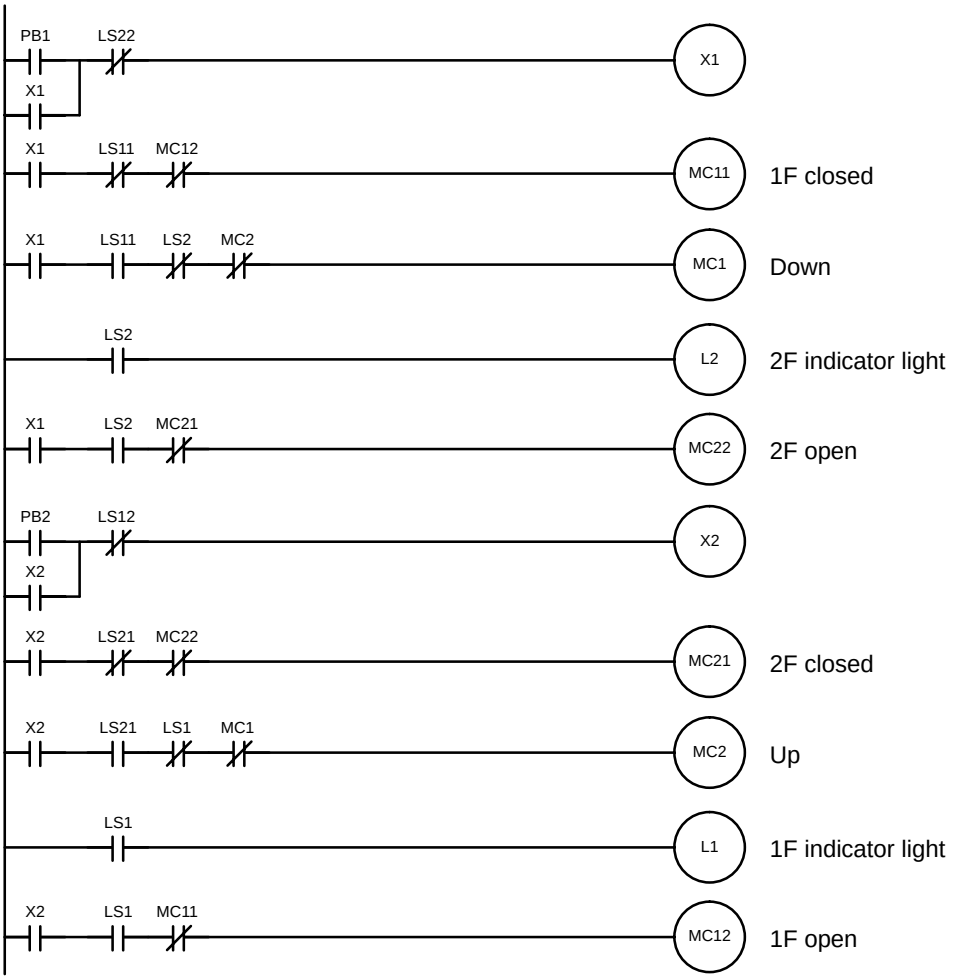
When food is placed on the lift and the “up” button is pressed to turn ON PB1, MC11 turns ON and the door closes. After the door has closed, LS11 turns ON. MC1 turns ON and the lift begins to go up. L1 turns OFF and the first floor indicator light turns off.

When the lift reaches the second floor (the customer serving area), L2 turns ON, MC1 turns OFF and the lift stops. L2 turns ON and the second-floor indicator light turns on. MC22 turns ON and the second-floor door opens.

When the food is taken from the lift and the “down” button is pressed, PB2 turns ON. MC21 turns ON and the door closes. After the door has closed, LS21 turns ON. MC2 turns ON and the lift begins to go down. L2 turns OFF and the second-floor indicator light turns off.

When the lift reaches the first floor again, LS1 turns ON. MC2 turns OFF and the lift stops. L1 turns ON and the first-floor indicator light turns ON. MC12 turns ON and the door opens.

Sequence Program Diagram



I/O Allocation Bit addresses are assigned as follows for inputs and outputs.

Input

Wd	000
00	LS11
01	PB1
02	LS1
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

Wd	001
00	LS21
01	PB2
02	LS2
03	LS22
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

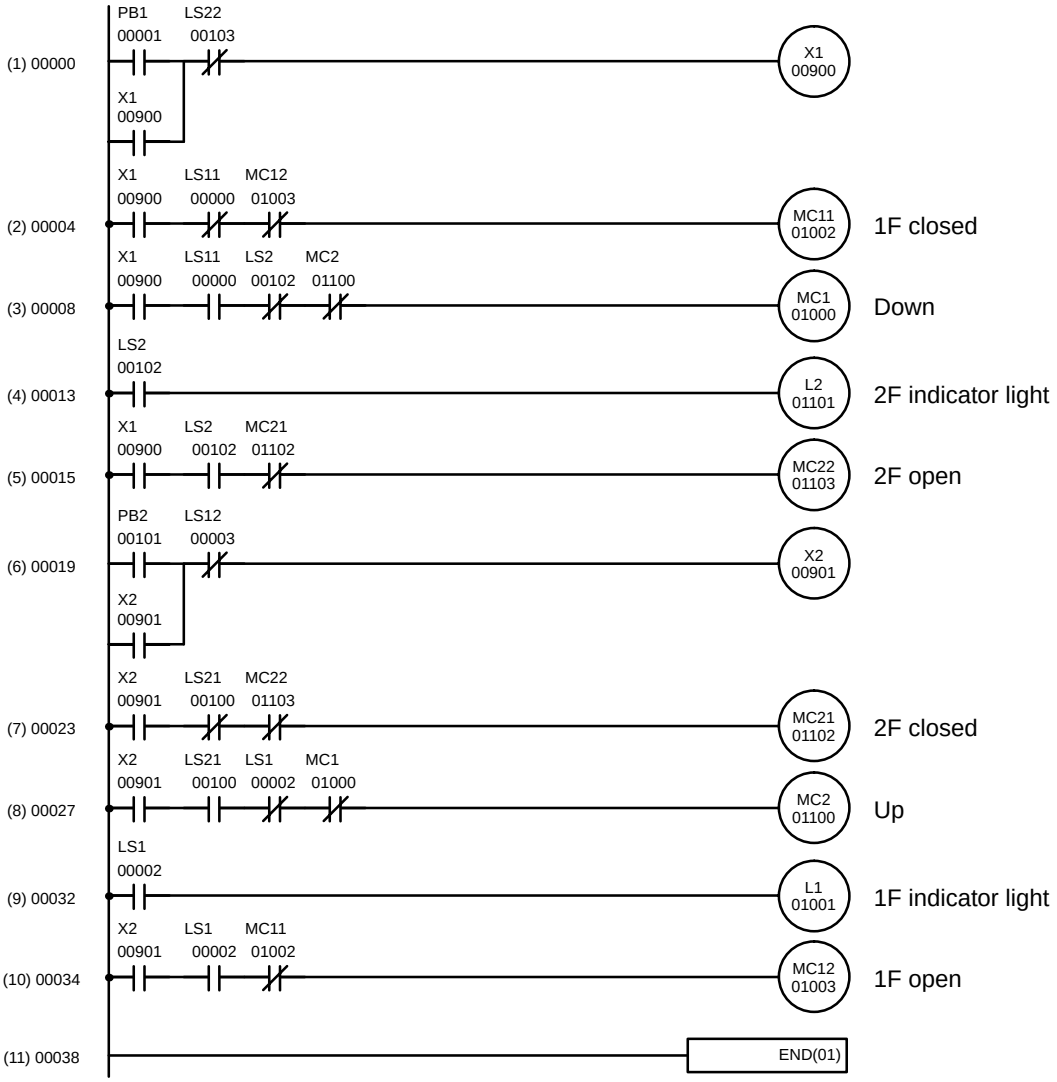
Output

Wd	010
00	MC1
01	L1
02	MC11
03	MC12
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

Wd	011
00	MC2
01	L2
02	MC21
03	MC22
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

Ladder Program

Following the sequence program and I/O allocation, create the ladder program.



Mnemonic Coding List

The following table shows the mnemonic coding for the ladder program. The subsequent pages explain the procedure for entering the program using the Programming Console.

Program reference	Step	Instruction	Data
(1)	00000	LD	00001
	00001	OR	00900
	00002	AND NOT	00103
	00003	OUT	00900
(2)	00004	LD	00900
	00005	AND NOT	00000
	00006	AND NOT	01003
	00007	OUT	01002
(3)	00008	LD	00900
	00009	AND	00000
	0010	AND NOT	00102
	00011	AND NOT	01100
	00012	OUT	01000
(4)	00013	LD	00102
	00014	OUT	01101

Program reference	Step	Instruction	Data
(5)	00015	LD	00900
	00016	AND	00102
	00017	AND NOT	01102
	00018	OUT	01103
(6)	00019	LD	00101
	00020	OR	00901
	00021	AND NOT	00003
	00022	OUT	00901
(7)	00023	LD	00901
	00024	AND NOT	00100
	00025	AND NOT	01103
	00026	OUT	01102
(8)	00027	LD	00901
	00028	AND	00100
	00029	AND NOT	00002
	00030	AND NOT	01000
	00031	OUT	01100
(9)	00032	LD	00002
	00033	OUT	01001
(10)	00034	LD	00901
	00035	AND	00001
	00036	AND NOT	01002
	00037	OUT	01003
(11)	00038	END (01)	---

5-2-4 Writing the Program

Use the following procedure to write the program, according to the coding sheet. The numbers in parentheses are in reference to the previous ladder program.

- **(1) Writing Steps 00000 to 00003**

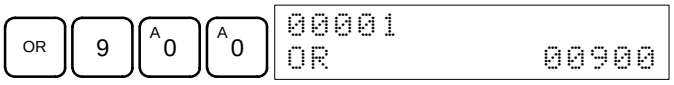
1, 2, 3...

1. Press the CLR Key to bring up the initial display.

2. Input the first instruction and bit address 00001. It is not necessary to input leading zeroes.

3. Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

4. Input OR and bit address 00900.



5. Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

6. Input AND, NOT and bit address 00103.

AND NOT B 1 A 0 D 3

```
00002
AND NOT     00103
```

7. Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

WRITE

```
00003READ
NOP (000)
```

8. Input the OUT instruction and bit address 00900.

OUT 9 A 0 A 0

```
00003
OUT         00900
```

9. Press the WRITE Key to write the instruction to Program Memory. The next program address will be displayed.

WRITE

```
00004READ
NOP (000)
```

• (2) to (11) Writing Steps 00004 to 00038

Enter steps 00004 to 00038 as follows:

(2) 00004 LD 9 A 0 A 0 WRITE

00005 AND NOT A 0 WRITE

00006 AND NOT B 1 A 0 A 0 D 3 WRITE

00007 OUT B 1 A 0 A 0 C 2 WRITE

(3) 00008 LD 9 A 0 A 0 WRITE

00009 AND A 0 WRITE

00010 AND NOT B 1 A 0 C 2 WRITE

00011 AND NOT B 1 B 1 A 0 A 0 WRITE

00012 OUT B 1 A 0 A 0 A 0 WRITE

(4) 00013 LD B 1 A 0 C 2 WRITE

00014 OUT B 1 B 1 A 0 B 1 WRITE

(5) 00015 LD 9 A 0 A 0 WRITE

00016 AND B 1 A 0 C 2 WRITE

00017 AND NOT B 1 B 1 A 0 C 2 WRITE

00018 OUT B 1 B 1 A 0 D 3 WRITE

(6) 00019 LD B 1 A 0 B 1 WRITE

00020 OR 9 A 0 B 1 WRITE

00021 AND NOT D 3 WRITE

00022 OUT 9 A 0 B 1 WRITE

(7) 00023 LD 9 A 0 B 1 WRITE

00024 AND NOT B 1 A 0 A 0 WRITE

00025 AND NOT B 1 B 1 A 0 D 3 WRITE

00026 OUT B 1 B 1 A 0 C 2 WRITE

(8) 00027 LD 9 A 0 B 1 WRITE

00028 AND B 1 A 0 A 0 WRITE

00029 AND NOT C 2 WRITE

00030 AND NOT B 1 A 0 A 0 A 0 WRITE

00031 OUT B 1 B 1 A 0 A 0 WRITE

(9) 00032 LD C 2 WRITE

00033 OUT B 1 A 0 A 0 B 1 WRITE

(10) 00034 LD 9 A 0 B 1 WRITE

00035 AND C 2 WRITE

00036 AND NOT B 1 A 0 A 0 C 2 WRITE

00037 OUT B 1 A 0 A 0 D 3 WRITE

(11) 00038 FUN A 0 A 0 B 1 WRITE

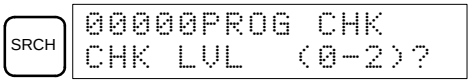
5-2-5 Checking the Program

This operation checks for programming errors and displays the program address and error when errors are found. It is possible in PROGRAM mode only.

RUN	MONITOR	PROGRAM
No	No	OK

1, 2, 3... 1. Press the CLR Key to bring up the initial display.

2. Press the SRCH Key. An input prompt will appear requesting the desired check level.



3. Input the desired check level (0, 1, or 2). The program check will begin when the check level is input, and the first error found will be displayed.

   Note Refer to *5-6 Programming Errors* for details on check levels and error displays.

4. Press the SRCH Key to continue the search. The next error will be displayed. Continue pressing the SRCH Key to continue the search.

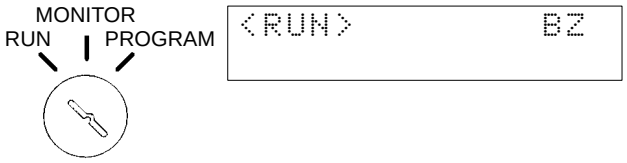
   A display like this will appear if an END instruction is reached without any errors being found:

If errors are displayed, edit the program to correct the errors and check the program again. Continue checking the program until all errors have been corrected.

5-3 Test Run

Run the SRM1 in MONITOR mode to check the program. Then follow the procedure described below to perform a test run.

1, 2, 3... 1. Change the mode switch to RUN mode.



2. Check the SRM1's LED indicator status. If the SRM1 is operating properly, the PWR, RUN, SD, and RD indicators should all be lit, the COMM indicator should flash, and all other indicators should be unlit. If the indicator status is anything other than this, refer to *5-4 Error Processing*.
3. Turn ON either PB1 or PB2 and check to see whether the operations are executed as described in *Explanation of Operations* in *5-2-3 Ladder Programming Example*. If the program is not executed, or the operations are different from those described in *Explanation of Operations*, check the program and the I/O wiring.

Note For checking the program, refer to *5-6 Programming Errors* and the *Programming Manual*.

5-4 Error Processing

The SRM1 is equipped with a variety of self-diagnosis functions to help identify and correct errors that might occur and reduce down time.

Errors are divided into two categories based on their severity. Fatal errors are more serious errors which stop SRM1 operation. Non-fatal errors are less serious and don't stop SRM1 operation.

5-4-1 Non-fatal Errors

SRM1 operation and program execution will continue after one or more of these errors have occurred. Although SRM1 operation will continue, the cause of the error should still be corrected and the error cleared as soon as possible.

When one of these errors occurs, the POWER and RUN indicators will remain lit and the ERR/ALM indicator will flash.

Message	FAL No.	Meaning and appropriate response
SYS FAIL FAL** (** is 01 to 99 or 9B.)	01 to 99	An FAL(06) instruction has been executed in the program. Check the FAL number to determine conditions that would cause execution, correct the cause, and clear the error.
	9B	An error has been detected in the PC Setup. Check flags AR 1300 to AR 1302, and correct as directed. AR 1300 ON: An incorrect setting was detected in the PC Setup (DM 6600 to DM 6614) when power was turned on. Correct the settings in PROGRAM Mode and turn on the power again. AR 1301 ON: An incorrect setting was detected in the PC Setup (DM 6615 to DM 6644) when switching to RUN Mode. Correct the settings in PROGRAM Mode and switch to RUN Mode again. AR 1302 ON: An incorrect setting was detected in the PC Setup (DM 6645 to DM 6655) during operation. Correct the settings and clear the error.
SCAN TIME OVER	F8	Watchdog timer has exceeded 100 ms. (SR 25309 will be ON.) This indicates that the program cycle time is longer than recommended. Reduce cycle time if possible. (The SRM1 can be set so that this error won't be detected.)
Communication Error (no message)	None	If an error occurs in CompoBus/S communications, the COMM indicator will be off and the ERC indicator will be lit. There is no error flag. Check the Slave and the connecting cables and restart.
		The COMM indicator will be OFF if an error occurs in communications through the Peripheral Port or RS-232C Port. AR 0804 will be ON if an error occurred in communications through the RS-232C port. AR 0812 will be ON if an error occurred in communications through the peripheral port. Check the connecting cables and restart.

5-4-2 Fatal Errors

SRM1 operation and program execution will stop and all outputs from the SRM1 will be turned OFF when any of these errors have occurred. SRM1 operation cannot be restarted until the SRM1 is turned off and then on again or the Programming Console is used to switch the SRM1 to PROGRAM mode and clear the fatal error.

All SRM1 indicators will be OFF for the power interruption error. For all other fatal operating errors, the POWER and ERR/ALM indicators will be lit. The RUN indicator will be OFF.

Message	FALS No.	Meaning and appropriate response
Power interruption (no message)	None	Power has been interrupted for at least 10 ms. Check power supply voltage and power lines. Try to power-up again.
MEMORY ERR	F1	AR 1308 ON: An unspecified bit area exists in the user program. Check the program and correct errors.
		AR 1309 ON: An error has occurred in the flash memory. Since the number of writings to the flash memory has exceeded the specified level, replace the SRM1.
		AR 1310 ON: A checksum error has occurred in read-only DM (DM 6144 to DM 6599). Check and correct the settings in the read-only DM area.
		AR 1311 ON: A checksum error has occurred in the PC Setup. Initialize all of the PC Setup and reinput.
		AR 1312 ON: A checksum error has occurred in the program. Check the program and correct any errors detected.
		AR 1314 ON: Power interruption hold area was not held. Clear the error and reset the settings of the power interruption hold area.
		AR 1315 ON: An error has occurred in CompoBus/S communications. If the error cannot be corrected, replace the SRM1.
NO END INST	F0	END(01) is not written in the program. Write END(01) at the end of the program.
SYS FAIL FALS** (** is 01 to 99 or 9F.)	01 to 99	A FALS(07) instruction has been executed in the program. Check the FALS number to determine the conditions that caused execution, correct the cause, and clear the error.
	9F	The cycle time has exceeded the FALS 9F Cycle Time Monitoring Time (DM 6618). Check the cycle time and adjust the Cycle Time Monitoring Time if necessary.

5-4-3 Identifying Errors

Errors can be identified from error messages displayed on the Programming Console, error flags in the SR or AR areas, and the error code in SR 25300 to SR 25307.

Error Messages Error messages generated by the self-diagnosis function can be read from a Programming Console or host computer running SYSMAC Support Software.

Error Flags When the self-diagnosis function detects a hardware error, it will turn on the corresponding error flags in the SR and AR areas.

Error Code When an error is detected by the self-diagnosis function, the corresponding error code is written to SR 25300 to SR 25307. (The error code is an 2-digit hexadecimal code.)

5-4-4 User-defined Errors

There are three instructions that the user can use to define his own errors or messages. FAL(06) causes a non-fatal error, FAL(07) causes a fatal error, and MSG(46) sends a message to the Programming Console or host computer connected to the SRM1.

FAILURE ALARM – FAL(06) FAL(06) is an instruction that causes a non-fatal error. The following will occur when an FAL(06) instruction is executed:

1, 2, 3...

1. The ERR indicator on the SRM1 will flash, but operation will continue.
2. The instruction's 2-digit BCD FAL number (01 to 99) will be written to SR 25300 to SR 25307.

The FAL numbers can be set arbitrarily to indicate particular conditions. The same number cannot be used as both an FAL number and an FALS number.

To clear an FAL error, correct the cause of the error and then execute FAL 00 or clear the error using the Programming Console.

SEVERE FAILURE ALARM – FALS(07)

FALS(07) is an instruction that causes a fatal error. The following will occur when an FALS(07) instruction is executed:

1, 2, 3...

1. Program execution will be stopped and outputs will be turned OFF.
2. The ERR indicator on the SRM1 will be lit.
3. The instruction's 2-digit BCD FALS number (01 to 99) will be written to SR 25300 to SR 25307.

The FALS numbers can be set arbitrarily to indicate particular conditions. The same number cannot be used as both an FAL number and an FALS number.

To clear an FALS error, switch the SRM1 to PROGRAM Mode, correct the cause of the error, and then clear the error using the Programming Console.

MESSAGE – MSG(46)

MSG(46) is used to display a message on the Programming Console. The message, which can be up to 16 characters long, is displayed when the instruction's execution condition is ON.

5-4-5 Reading/Clearing Error Messages

This operation is used to display and clear error messages. It is possible to display and clear non-fatal errors and MESSAGE instruction messages in any mode, but fatal errors can be displayed and cleared in PROGRAM mode only.

RUN	MONITOR	PROGRAM
OK	OK	OK

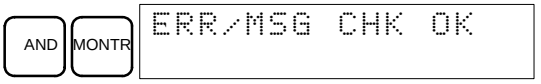
Before inputting a new program, any error messages recorded in memory should be cleared. It is assumed here that the causes of any of the errors for which error messages appear have already been taken care of. If the buzzer sounds when an attempt is made to clear an error message, eliminate the cause of the error, and then clear the error message. (Refer to *5-7 Troubleshooting Flowcharts* for troubleshooting information.)

Key Sequence

Follow the procedure below to display and clear messages.

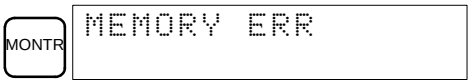
1, 2, 3...

1. Press the CLR Key to bring up the initial display.
2. Press the FUN and then the MONTR Key to begin the operation. If there are no messages, the following display will appear:

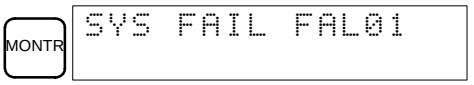


If there are messages, the most serious message will be displayed when the MONTR Key is pressed. Pressing MONTR again will clear the present message and display the next most serious error message. Continue pressing MONTR until all messages have been cleared. These are some examples of error messages:

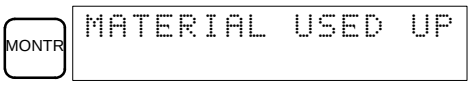
A memory error:



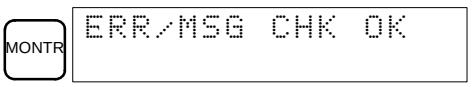
A system error:



A message:



All messages cleared:



⚠ **Caution** Check to be sure that no equipment is affected when turning the SRM1's power supply on or off, or when entering the password. Be careful not to cause any accidents when starting or stopping SRM1 operation.

5-5 Programming Console Operation Errors

The following error messages may appear when performing operations on the Programming Console. Correct the error as indicated and continue with the operation.

Message	Meaning and appropriate response
REPL ROM	An attempt was made to write to write-protected memory. Set bits 00 to 03 of DM 6602 to "0."
PROG	The instruction at the last address in memory is not NOP(00). Erase all unnecessary instructions at the end of the program.
ADDR OVER	An address was set that is larger than the highest memory address in Program Memory. Input a smaller address.
SETDATA ERR	FALS 00 has been input, and "00" cannot be input. Reinput the data.
I/O NO. ERR	A data area address has been designated that exceeds the limit of the data area, e.g., an address is too large. Confirm the requirements for the instruction and re-enter the address.

5-6 Programming Errors

These errors in program syntax will be detected when the program is checked using the Program Check operation.

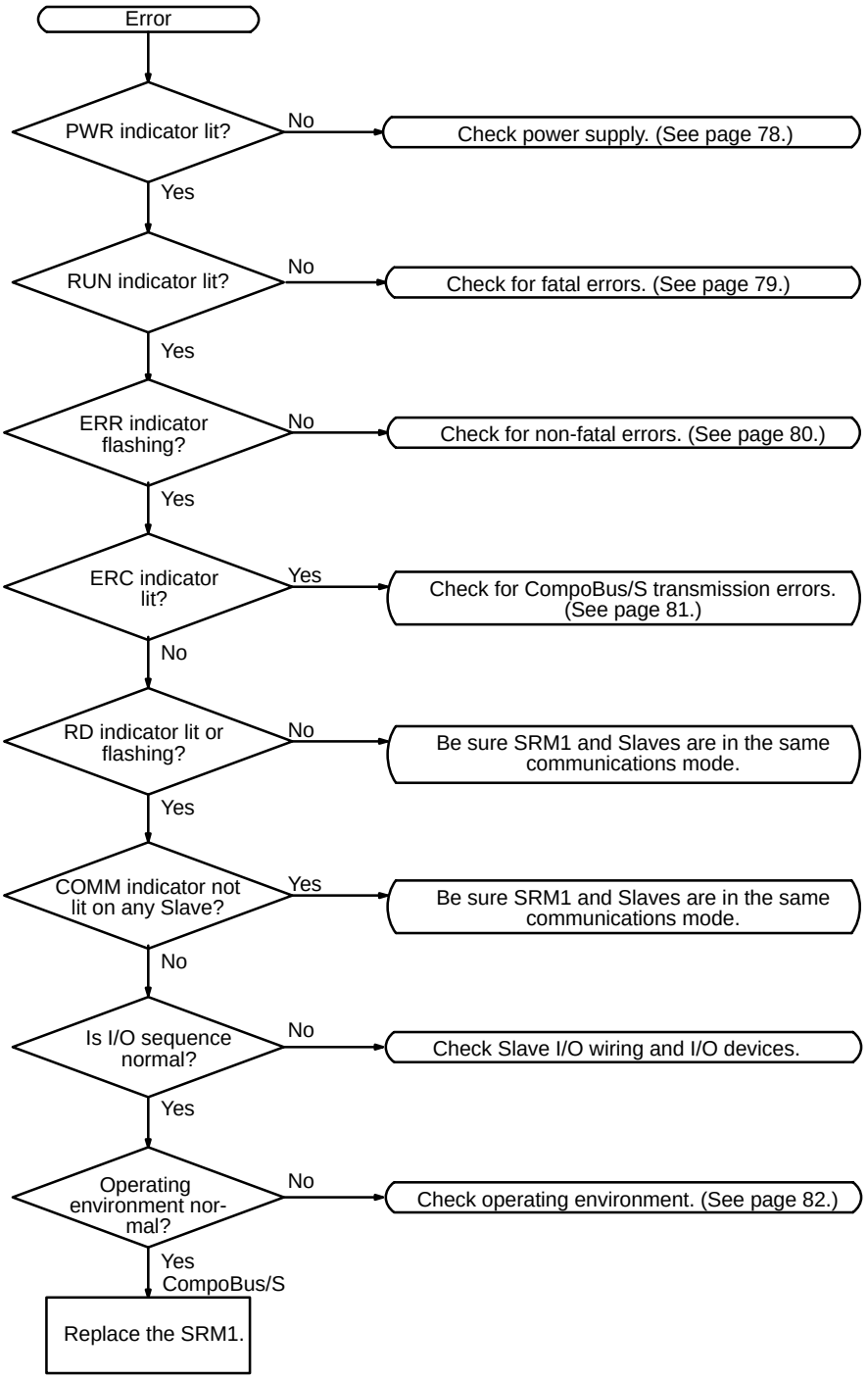
Three levels of program checking are available. The desired level must be designated to indicate the type of errors that are to be detected. The following table provides the error types, displays, and explanations of all syntax errors. Check level 0 checks for type A, B, and C errors; check level 1, for type A and B errors; and check level 2, for type A errors only.

Type	Message	Meaning and appropriate response
A	?????	The program has been damaged, creating a non-existent function code. Re-enter the program.
	CIRCUIT ERR	The number of logic blocks and logic block instructions does not agree, i.e., either LD or LD NOT has been used to start a logic block whose execution condition has not been used by another instruction, or a logic block instruction has been used that does not have the required number of logic blocks. Check your program.
	OPERAND ERR	A constant entered for the instruction is not within defined values. Change the constant so that it lies within the proper range.
	NO END INSTR	There is no END(01) in the program. Write END(01) at the final address in the program.
	LOCN ERR	An instruction is in the wrong place in the program. Check instruction requirements and correct the program.
	JME UNDEFD	A JME(05) instruction is missing for a JMP(04) instruction. Correct the jump number or insert the proper JME(04) instruction.
	DUPL	The same jump number or subroutine number has been used twice. Correct the program so that the same number is only used once for each.
	SBN UNDEFD	The SBS(91) instruction has been programmed for a subroutine number that does not exist. Correct the subroutine number or program the required subroutine.
	STEP ERR	STEP(08) with a section number and STEP(08) without a section number have been used incorrectly. Check STEP(08) programming requirements and correct the program.
B	IL-ILC ERR	IL(02) and ILC(03) are not used in pairs. Correct the program so that each IL(02) has a unique ILC(03). Although this error message will appear if more than one IL(02) is used with the same ILC(03), the program will executed as written. Make sure your program is written as desired before proceeding.
	JMP-JME ERR	JMP(04) and JME(05) are not used in pairs. Make sure your program is written as desired before proceeding.
	SBN-RET ERR	If the displayed address is that of SBN(92), two different subroutines have been defined with the same subroutine number. Change one of the subroutine numbers or delete one of the subroutines. If the displayed address is that of RET(93), RET(93) has not been used properly. Check requirements for RET(93) and correct the program.
C	COIL DUPL	The same bit is being controlled (i.e., turned ON and/or OFF) by more than one instruction (e.g., OUT, OUT NOT, DIFU(13), DIFD(14), KEEP(11), SFT(10)). Although this is allowed for certain instructions, check instruction requirements to confirm that the program is correct or rewrite the program so that each bit is controlled by only one instruction.
	JMP UNDEFD	JME(05) has been used with no JMP(04) with the same jump number. Add a JMP(04) with the same number or delete the JME(05) that is not being used.
	SBS UNDEFD	A subroutine exists that is not called by SBS(91). Program a subroutine call in the proper place, or delete the subroutine if it is not required.

5-7 Troubleshooting Flowcharts

Use the following flowcharts to troubleshoot errors that occur during operation.

Main Check

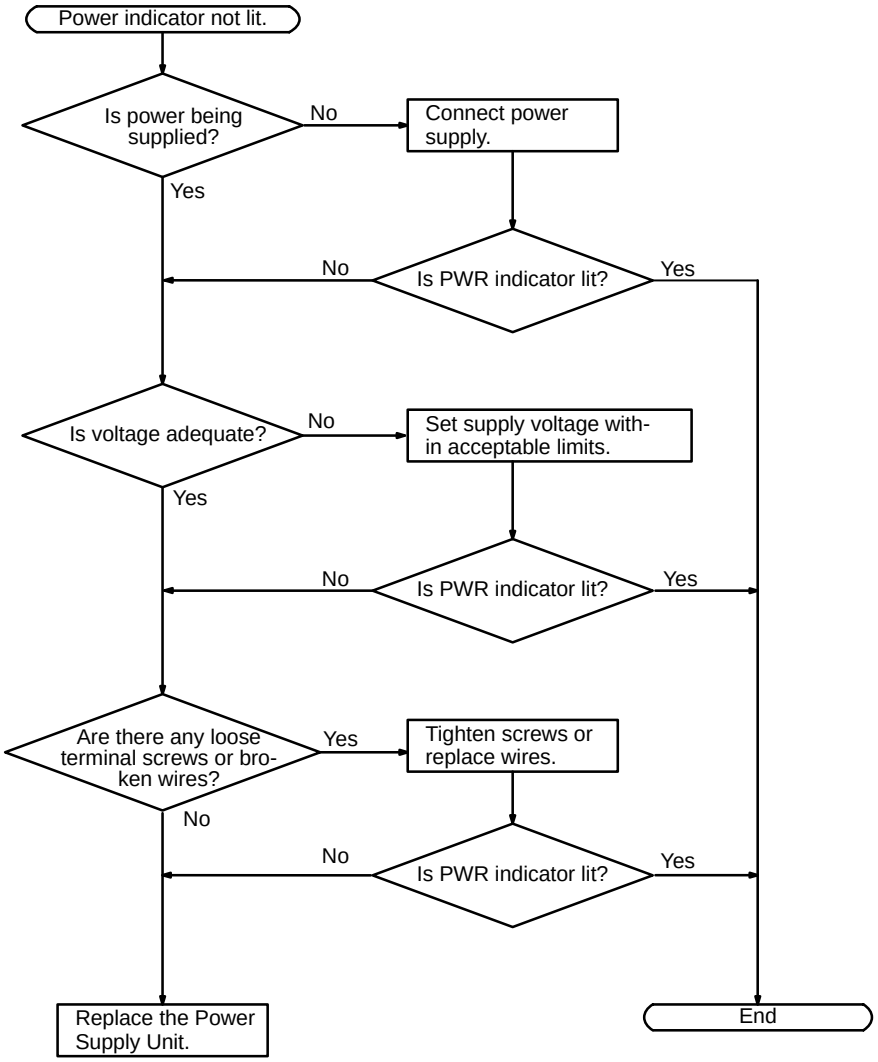


If the LED indicators are all normal (i.e., ERC off, SD lit, and RD lit), yet there is a communication error, check the following points. (For details on the Slaves, refer to the *CompoBus/S Operation Manual (W266)*.)

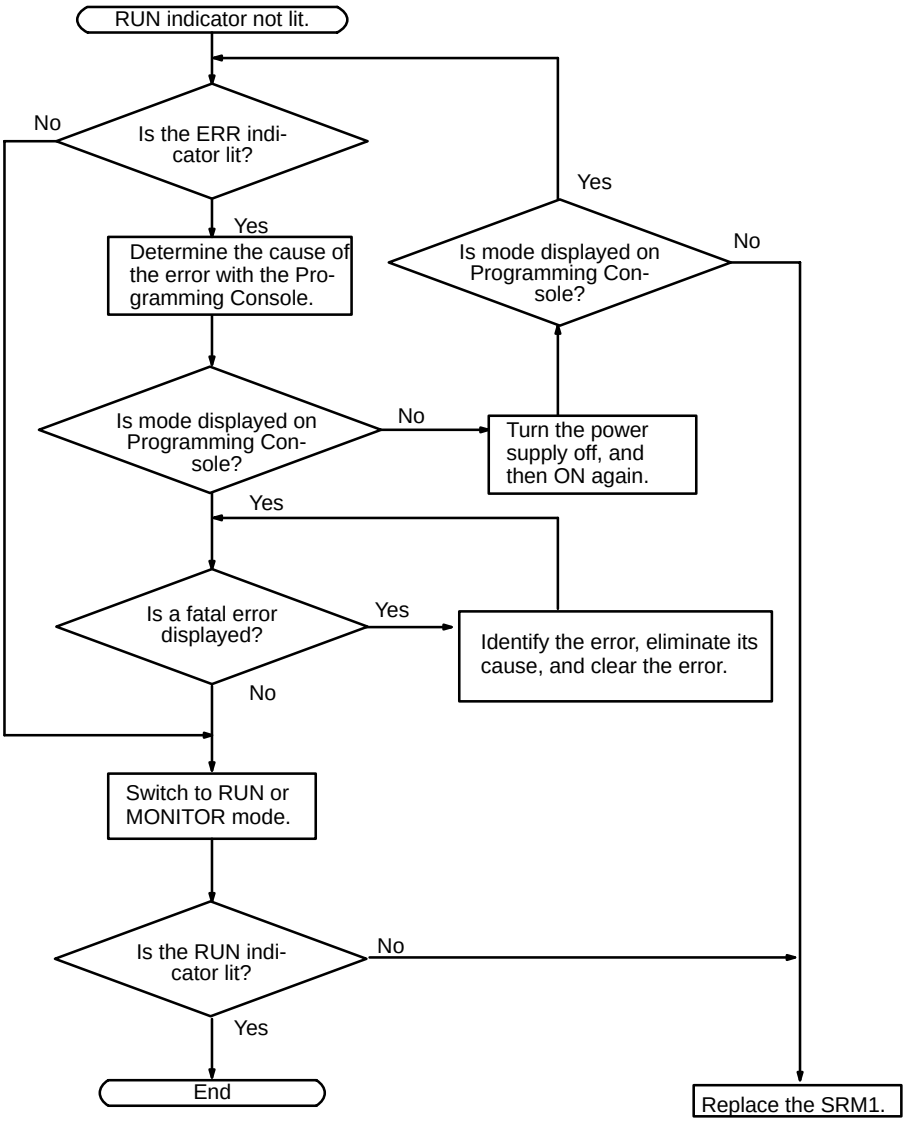
- Are the Slave's I/O power supply and I/O wiring okay?
- Are the Slave's power supply and address settings okay? Is the Slave itself operating properly?
- Is the termination correctly connected to the end of the transmission line (i.e. the end farthest from the SRM1)?

- Are the lengths of the main line and the branch lines, and the total length, all within the prescribed limits?
- Are flat cables and VCTF cables mixed together among the transmission lines?

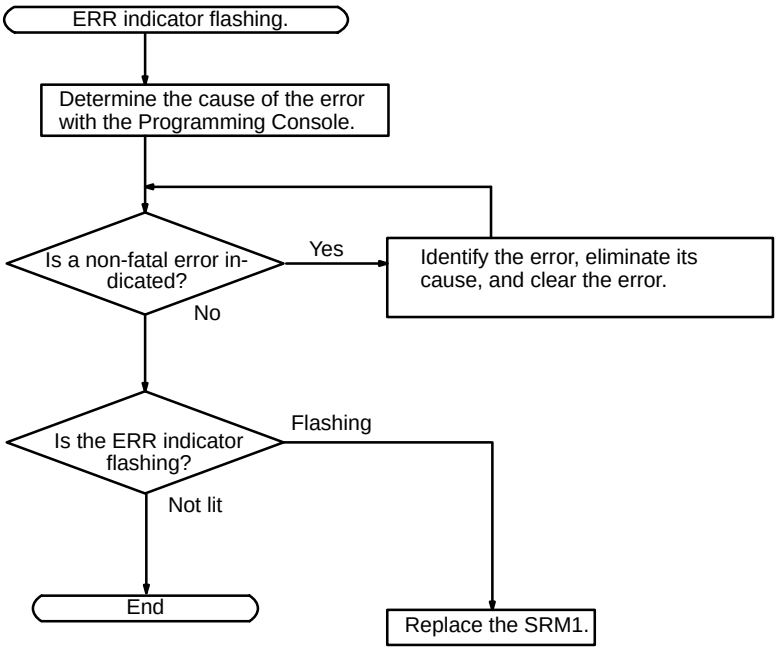
Power Supply Check



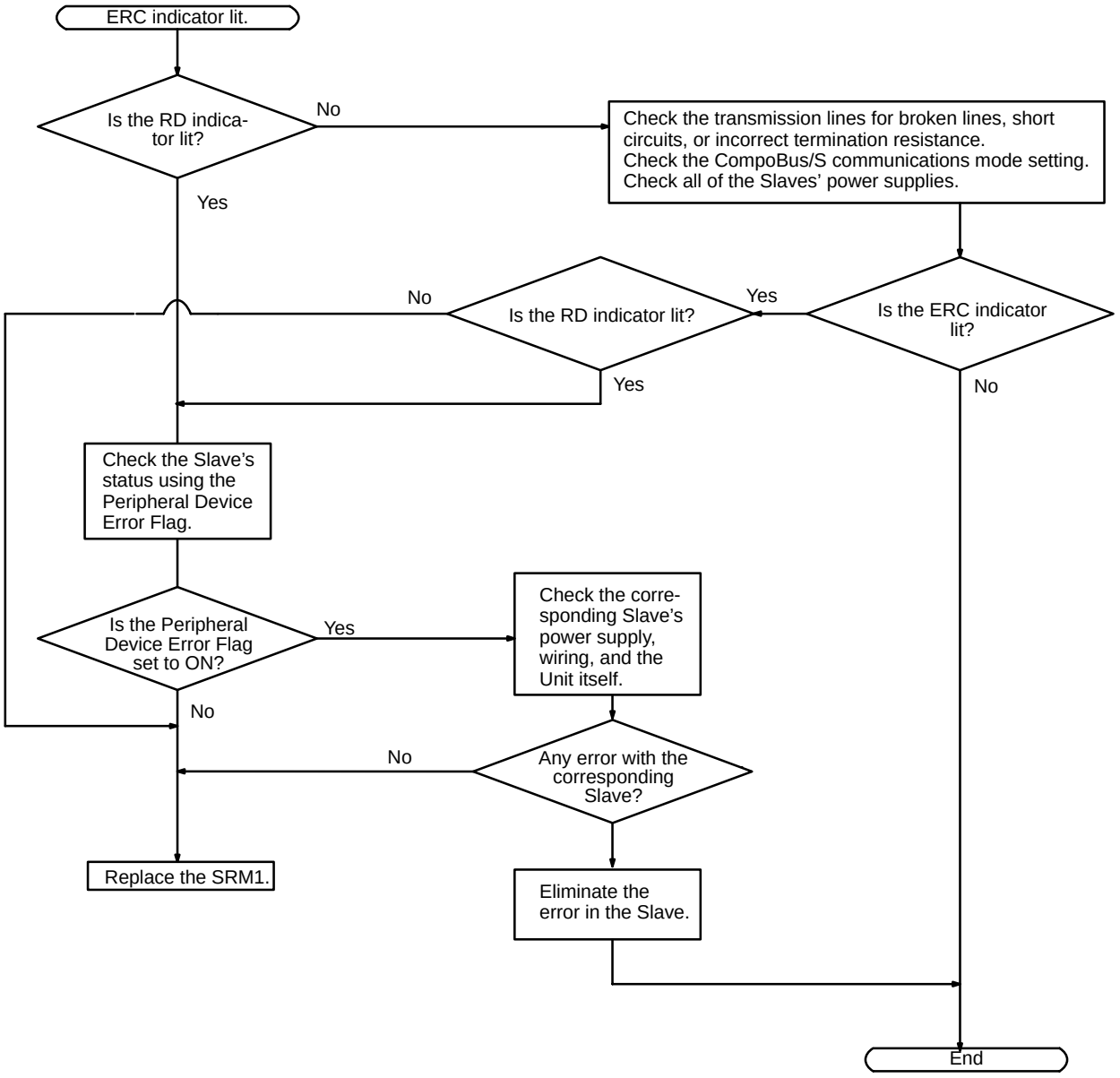
Fatal Error Check



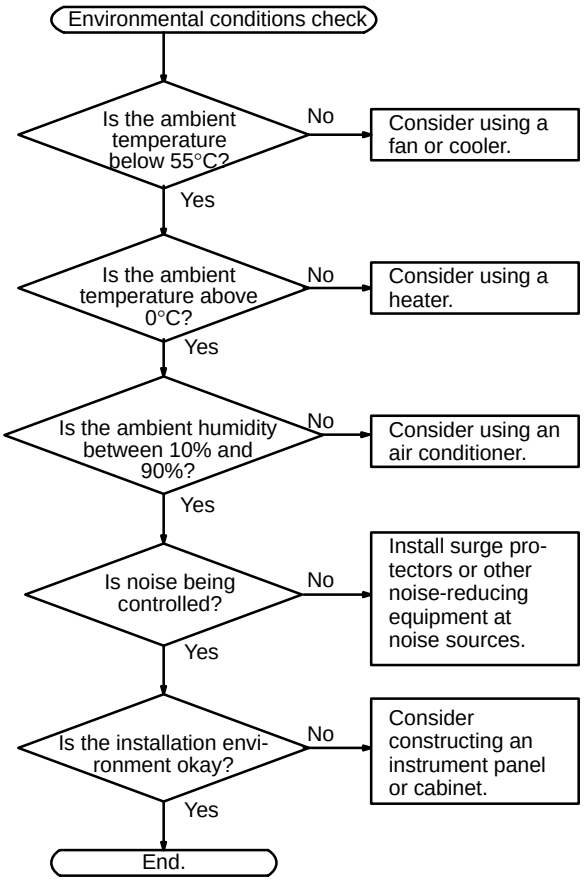
Non-fatal Error Check



CompoBus/S Transmission Error Check



Environmental Conditions Check



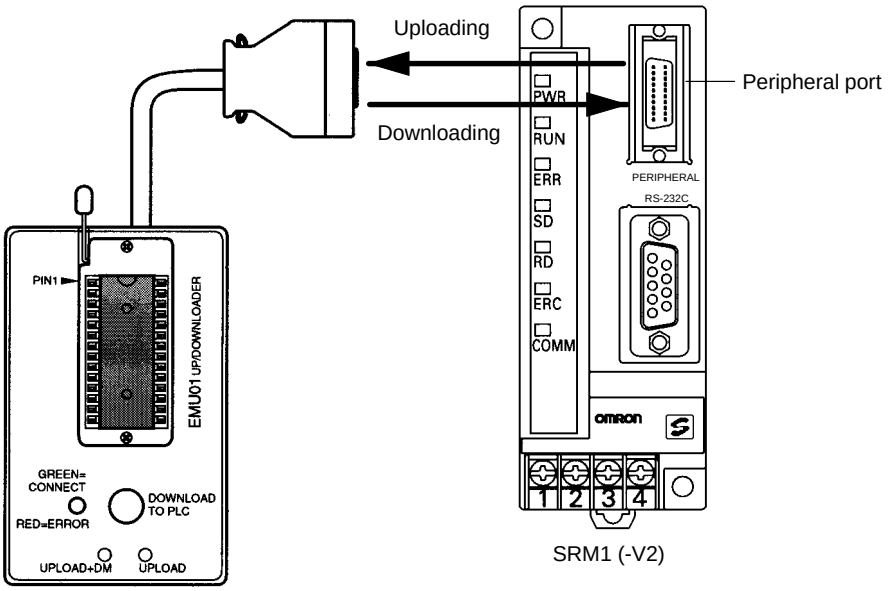
SECTION 6
Expansion Memory Unit

This section describes how to use the CPM1-EMU01-V1 Expansion Memory Unit. Follow the handling precautions and procedures to properly use the Unit.

6-1 Overview 84
6-1-1 Memory Areas 84
6-1-2 Precautions 84
6-2 Specifications and Nomenclature 85
6-2-1 Specifications 85
6-2-2 Nomenclature 85
6-3 Handling 86
6-3-1 Mounting/Removing EEPROM 86
6-3-2 PC Connections 87
6-3-3 Uploading Programs 88
6-3-4 Downloading Programs 89

6-1 Overview

The CPM1-EMU01-V1 Expansion Memory Unit is a program loader for small-size or micro PCs. Using the CPM1-EMU01-V1, simple on-site transfer of user programs and data memory between PCs is possible.



SRM1 (-V2)

Note The “PLC” in the “DOWNLOAD TO PLC” Button indicates PCs (Programmable Controllers).

6-1-1 Memory Areas

The memory areas that are uploaded or downloaded vary with the button used as shown in the following table.

Button	UPLOAD + DM	UPLOAD	DOWNLOAD TO PLC
Ladder program and expansion instructions	Read from PC to EEPROM.	Read from PC to EEPROM.	All contents of EEPROM written to PC.
DM 6144 to 6655		Not affected.	

Note For details on program size, DM area, and the availability of expansion instructions, refer to the relevant PC manual.

6-1-2 Precautions

- Do not attempt to use the CPM1-EMU01-V1 for any applications other than those described here. Doing so may result in malfunction.
- Do not attempt to upload or download data between different types of PC. Doing so may result in malfunction.
- Do not download when the PC is in RUN or MONITOR mode. If downloading is performed when the PC is running, it will automatically switch to PROGRAM mode and operation will stop.
- Do not attempt to disassemble, repair, or modify the CPM1-EMU01-V1. Any attempt to do so may result in malfunction, fire, or electric shock.
- After downloading has been completed, be sure to confirm the user program, data, and expansion instruction information. Not doing so may result in faulty operation.

- Before touching the EEPROM or the CPM1-EMU01-V1, first touch a grounded metallic object to discharge any static build-up. Not doing so may result in malfunction or damage.

6-2 Specifications and Nomenclature

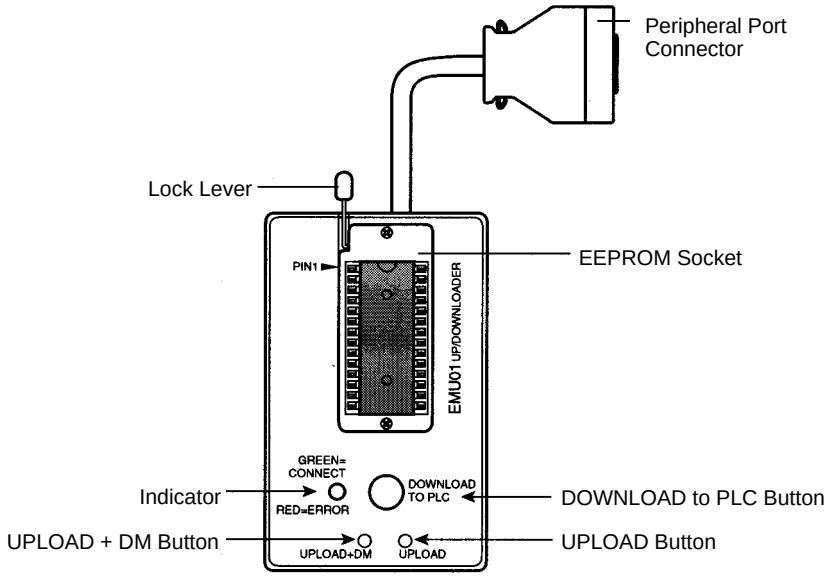
6-2-1 Specifications

Item	Specifications
Supported PCs	CPM1, CPM1A, CPM2A, CPM2C, SRM1 (-V2), CQM1, CQM1H
Read/Write memory areas	User program: 15.2 Kwords max. Data memory: DM 6144 to DM 6655 (Read-only DM and PC Setup) Expansion instructions: 18 instructions
Connector	Connector compatible with CPM1, CPM1A, CPM2A, SRM1 (-V2), and CQM1 PCs. For CPM2C and CQM1H PCs, connect via CS1W-CN114 or CPM2C-CN111 Connecting Cable.
Communications setting	1 start bit, 7 data bits, even parity, 2 stop bits, 9,600 bps
EEPROM (See note 1.)	256-Kbit EEPROM ATMEL: AT28C256 OMRON: EEROM-JD
Current consumption	129 mA max.
Dimensions	Main body (not including cables or connectors): 57 × 92 × 38 mm (W × H × D)
Weight	200 g max. (not including EEPROM)

Note 1. The EEPROM must be purchased separately.

2. For general specifications, refer to the relevant PC manual.

6-2-2 Nomenclature



Lock Lever
For mounting and removing EEPROM.

DOWNLOAD TO PLC Button
Writes all EEPROM data (ladder programs, data memory etc.) to the PC.

UPLOAD + DM Button
Reads PC user program and contents of DM 6144 to DM 6655 to EEPROM.

UPLOAD Button
Reads only PC user program to EEPROM.

Note The “PLC” in the “DOWNLOAD TO PLC” Button indicates PCs (Programmable Controllers).

LED Indicator

CONNECT (green)	ERR (red)	Meaning
OFF	OFF	Not connected to PC (power supply OFF).
ON	OFF	Connected to a recognized PC.
Blinking	OFF	Uploading/downloading data.
ON	Blinking	Host link communications error, retry by user.
OFF	ON	PC model and EEPROM data not compatible.
OFF	Blinking	One of the following errors has occurred: An unrecognized PC is connected. An EEPROM error (EEPROM not present, EEPROM defect, or no program to download) or checksum error.

6-3 Handling

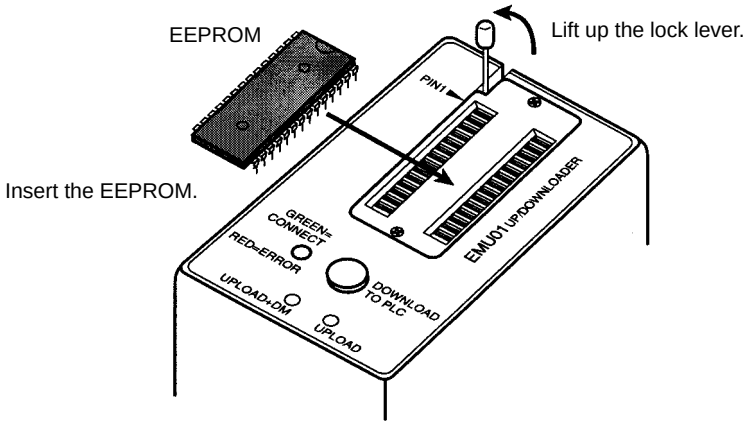
6-3-1 Mounting/Removing EEPROM

⚠ Caution Do not mount or remove the EEPROM with the CPM1-EMU01-V1 connected to the PC. Doing so may damage the EEPROM.

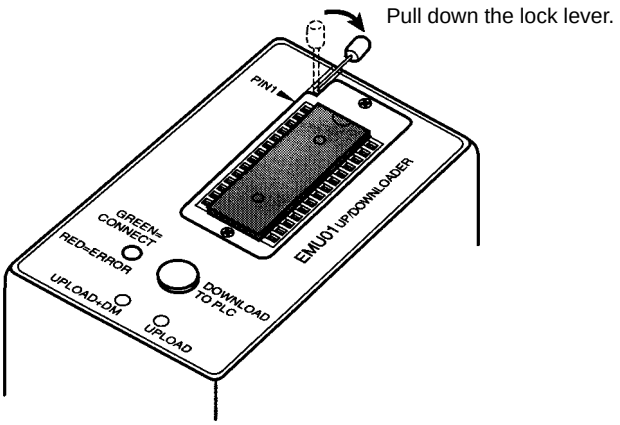
Mounting EEPROM

1, 2, 3...

1. Lift up the lock lever.
2. Straighten the pins on the EEPROM, line up with the socket and lower into the socket, as shown in the following diagram. If the EEPROM is loose, place it in the center of the socket.



3. Gently hold down the EEPROM and pull down the lock lever.



Removing EEPROM Lift up the lock lever and detach the EEPROM.

6-3-2 PC Connections

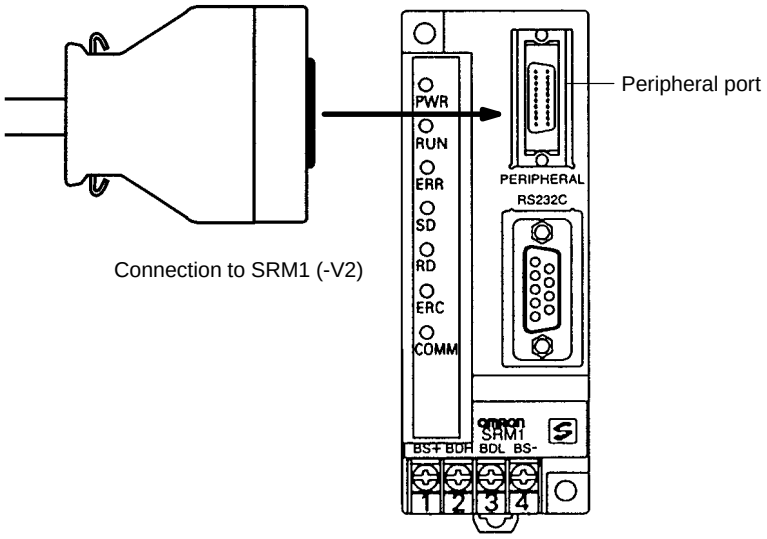
⚠ Caution Mount the EEPROM to the CPM1-EMU01-V1 before connecting the CPM1-EMU01-V1 to the PC.

⚠ Caution Do not disconnect the CPM1-EMU01-V1 from the PC when the indicator is blinking green.

CPM1, CPM1A, CPM2A, CQM1, and SRM1 (-V2) PCs

When connecting to the CPM1, CPM1A, CPM2A, CQM1 or SRM1 (-V2),insert the connector into the peripheral port making sure that the connector is oriented correctly.

- Insert the connector until it securely locks into place.
- Connections are not possible to the RS-232C port or any other port.



Connection to SRM1 (-V2)

CPM2C and CQM1H PCs

When connecting to the CPM2C or CQM1H, connect to the peripheral port via the CPM2C-CN111 or CS1W-CN114 Connecting Cable. Also, set the pins on the CPU Unit's DIP switch as follows:

CPM2C	Pin 1: ON (see note)
	Pin 2: ON
CQM1H	Pin 5: ON (see note)
	Pin 7: ON

Note If pin 1 on the CPM2C or pin 5 on the CQM1H is OFF, connection is still possible if the peripheral port is set to the defaults.

Peripheral Port Communications Settings

The peripheral port must be set to the default communications settings shown below.

Start bit: 1

Data bits: 7

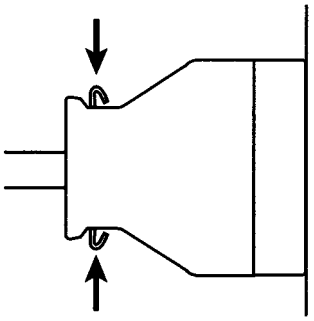
Stop bits: 2

Parity: Even

Baud rate: 9,600 bps

Disconnecting

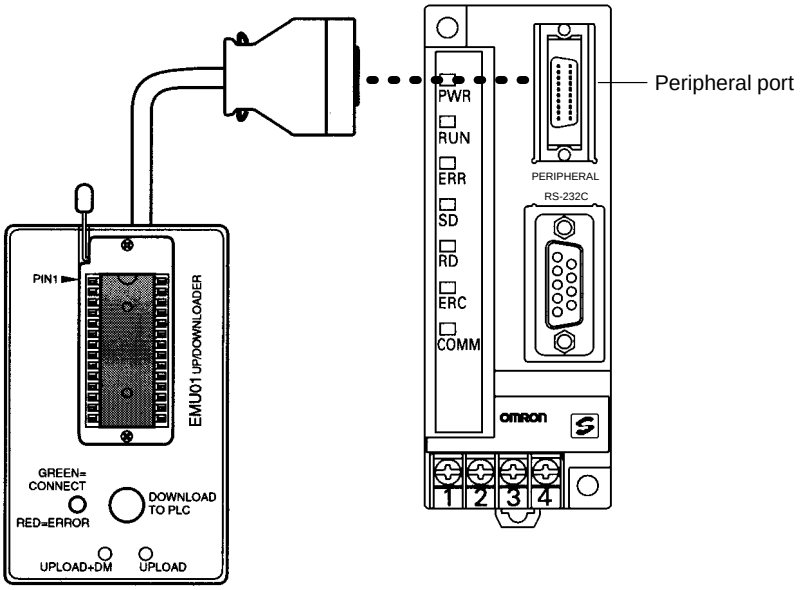
Press the levers on the top and bottom of the connector inwards to unlock the connector and pull out as shown in the following diagram.



Note Do not attempt to remove the connector by pulling the cable.

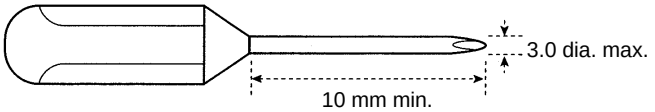
6-3-3 Uploading Programs

Ladder programs and the contents of data memory can be uploaded to the EEPROM using the following procedure. The buttons used will determine whether the contents of data memory are uploaded or not.



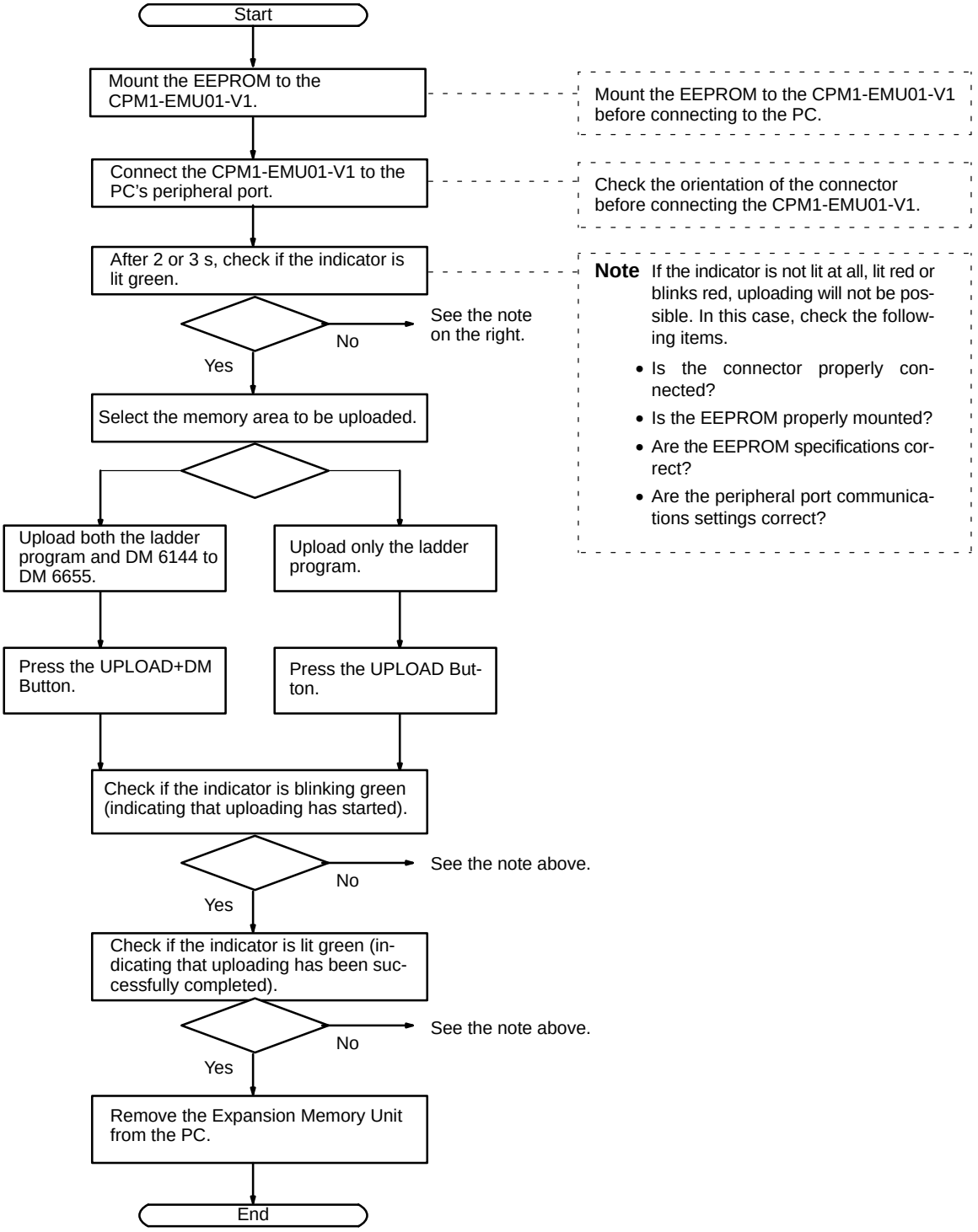
Button	UPLOAD + DM	UPLOAD
Ladder program and expansion instructions	Read from PC to EEPROM.	Read from PC to EEPROM.
DM6144 to 6655		Not affected.

Note Use a Phillips screwdriver or other tool with a diameter of 3.0 mm max. and a blade length of 10 mm min. to press the upload button.



Uploading is possible even if the PC is in RUN or MONITOR mode.

Operation Procedure



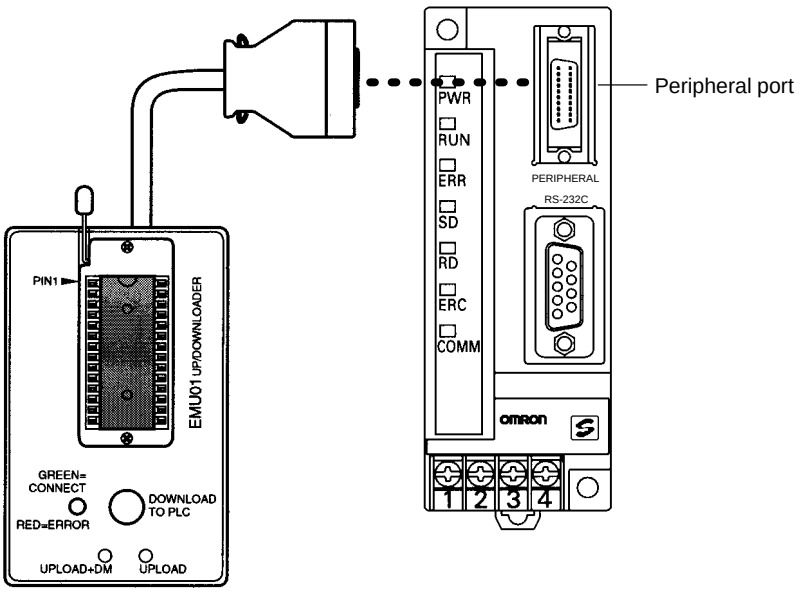
6-3-4 Downloading Programs

Ladder programs and the contents of data memory can be downloaded from the EEPROM to the PC using the procedure given below. When downloading, note the following points.

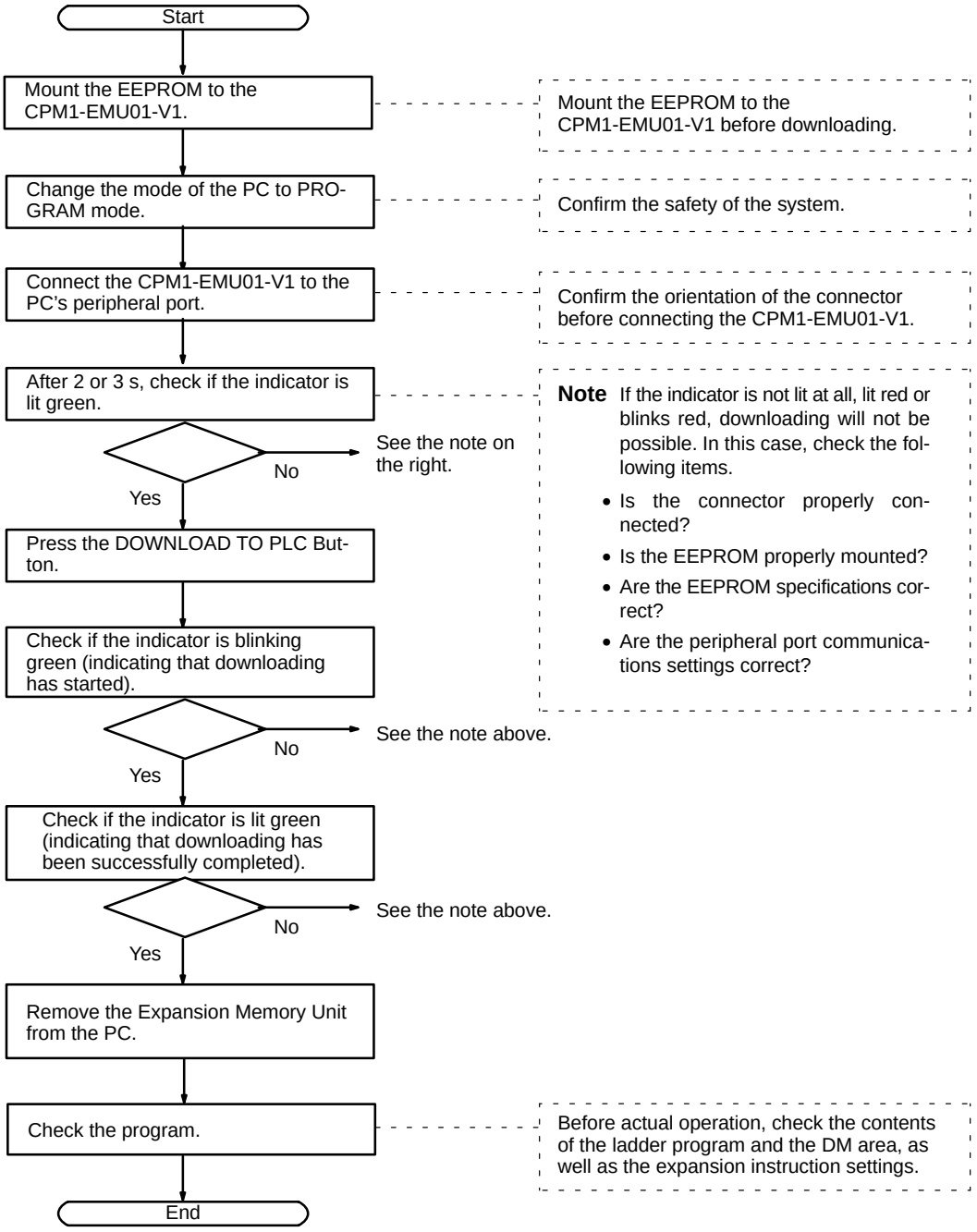
⚠ Caution If the PC is in RUN or MONITOR mode when downloading is started, it will automatically switch to PROGRAM mode and operation will stop. Confirm that no adverse effects will occur to the system before downloading. Not doing so may result in unexpected operation.

Note

1. If the PC is in RUN or MONITOR mode, switch the PC mode to PROGRAM mode.
2. Confirm that the program or other data to be downloaded to the PC is compatible with the PC before downloading.
3. For the CQM1 or CQM1H, when downloading programs that use user-assigned function codes for expansion instructions, be sure that pin 4 on the CPU Unit's DIP switch is ON. If these programs are downloaded while the pin is OFF, the assignments for the expansion instructions will return to their default settings.
4. After downloading has been completed, be sure to confirm the user program, data, and expansion instructions.
5. When the DOWNLOAD TO PLC Button is pressed, all EEPROM data (ladder programs, data memory etc.) is written to the PC.



Operation Procedure



Appendix A
Standard Models

There are three possible power supply configurations.

- Network Power Supply:
  Power is provided from a single source over a network, using special-purpose flat cable.
- Multiple Power Supply:
  Power must be provided separately for communications and I/O. The power supply for communications can be provided via special-purpose flat cable, but the power supply for I/O cannot.
- Local Power Supply:
  Power can be provided from a single source, which can be a source other than the special-purpose flat cable.

SRT1-series Remote Terminals and Sensor Terminals can operate in high-speed communications mode only. Be sure to use SRT2-series Remote Terminals and Sensor Terminals when the SRM1(-V2) is used in long-distance communications mode.

The Analog I/O Terminals can be used with SRM1(-V2) PCs only.

Remote Terminals

SRT2 Series High-speed or long-distance communications	SRT1 Series High-speed communications only	Specifications
SRT2-ID04	SRT1-ID04	4 transistor inputs, multiple power supply, NPN
SRT2-ID04-1	SRT1-ID04-1	4 transistor inputs, multiple power supply, PNP
SRT2-ID08	SRT1-ID08	8 transistor inputs, multiple power supply, NPN
SRT2-ID08-1	SRT1-ID08-1	8 transistor inputs, multiple power supply, PNP
SRT2-ID16	SRT1-ID16	16 transistor inputs, multiple power supply, NPN
SRT2-ID16-1	SRT1-ID16-1	16 transistor inputs, multiple power supply, PNP
SRT2-ID16T	None	16 transistor inputs, multipoint common terminal, multiple power supply, NPN
SRT2-ID16T-1		16 transistor inputs, multipoint common terminal, multiple power supply, PNP
SRT2-OD04	SRT1-OD04	4 transistor outputs, multiple power supply, NPN
SRT2-OD04-1	SRT1-OD04-1	4 transistor outputs, multiple power supply, PNP
SRT2-OD08	SRT1-OD08	8 transistor outputs, multiple power supply, NPN
SRT2-OD08-1	SRT1-OD08-1	8 transistor outputs, multiple power supply, PNP
SRT2-OD16	SRT1-OD16	16 transistor outputs, multiple power supply, NPN
SRT2-OD16-1	SRT1-OD16-1	16 transistor outputs, multiple power supply, PNP
SRT2-OD16T	None	4 transistor outputs, multiple power supply, NPN
SRT2-OD16T-1		16 transistor outputs, multiple power supply, PNP
SRT2-ROC08	SRT1-ROC08	8 relay outputs, local power supply
SRT2-ROC16	SRT1-ROC16	16 relay outputs, local power supply
SRT2-ROF08	SRT1-ROF08	8 power MOSFET relay outputs, local power supply
SRT2-ROF16	SRT1-ROF16	16 power MOSFET relay outputs, local power supply
SRT2-MD16T	None	8 transistor inputs/8 transistor outputs, multipoint common terminal, multiple power supply, NPN
SRT2-MD16T-1		8 transistor inputs/8 transistor outputs, multiple power supply, PNP

Connector Terminals

Model	Specifications
SRT2-VID08S	8 transistor inputs, sensor cable connector, multiple power supply, NPN
SRT2-VID08S-1	8 transistor inputs, sensor cable connector, multiple power supply, PNP
SRT2-VID16ML	16 transistor inputs, MIL connector, multiple power supply, NPN
SRT2-VID16ML-1	16 transistor inputs, MIL connector, multiple power supply, PNP
SRT2-VOD08S	8 transistor outputs, sensor cable connector, multiple power supply, NPN
SRT2-VOD08S-1	8 transistor outputs, sensor cable connector, multiple power supply, PNP
SRT2-VOD16ML	16 transistor outputs, MIL connector, multiple power supply, NPN
SRT2-VOD16ML-1	16 transistor outputs, MIL connector, multiple power supply, PNP

Note All the above models support both high-speed and long-distance communications modes.

Remote I/O Terminals

Model	Specifications
SRT1-ROF08	16 inputs, NPN, PCB-mounting type
SRT1-ROF16	16 outputs, NPN, PCB-mounting type

Note The above models do not support long-distance communications mode.

Sensor Amplifier Terminals

SRT2 Series High-speed or long-distance communications	SRT1 Series High-speed communications only	Specifications
SRT2-TID04S (See note.)	SRT1-TID04S	4 inputs (1 word x 4), network power supply
SRT2-TKD04S (See note.)	SRT1-TKD04S	4 inputs (4 words x 1), network power supply
SRT2-XID04S (See note.)	SRT1-XID04S	4 inputs (1 word x 4), Expansion Sensor Amplifier Terminal
SRT2-XKD04S (See note.)	SRT1-XKD04S	4 inputs (4 words x 1), Expansion Sensor Amplifier Terminal

Note To be marketed in the near future.

Analog I/O Terminals

Model	Specifications
SRT2-AD04	4 analog inputs (settable to 1, 2, 3, or 4 inputs), network power supply
SRT2-DA02	2 analog outputs (settable to 1 or 2 outputs), network power supply

Note All the above models support both high-speed and long-distance communications modes.

Sensor Terminals

SRT2 Series High-speed or long-distance communications	SRT1 Series High-speed communications only	Specifications
SRT2-ID08S (See note.)	SRT1-ID08S	8 inputs, network power supply
SRT2-OD08S (See note.)	SRT1-OD08S	8 outputs, local power supply
SRT2-MD08S (See note.)	SRT1-MD08S	4 inputs, 4 outputs, network power supply

Note To be marketed in the near future.

Bit Chain Terminal

Model	Specifications
SRT1-B1T	8 inputs/outputs (I/O set via switch), local power supply

Note The above model does not support long-distance communications mode.

CPM1A/CPM2A I/O Link Unit

Model	Specifications
CPM1A-SRT21	8 inputs, 8 outputs Exchanges data with CPM1A/CPM2A CPU Unit.

Note All the above models support both high-speed and long-distance communications modes.

Connection Devices

Communications Cables

Model	Specifications
Commercially available	VCTF cable (JIS C3306), 0.75 mm^2 x 2 conductors
SCA1-4F10	Special Flat Cable, 100 m, 0.75 mm^2 x 4 conductors

Specified Communications Cables

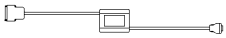
Model	Manufacturer	Comments
#9409	Belden	USA manufacturer

Note The electrical characteristics of the above cable are the same as those of the following: VCTF cable (JIS C3306), 0.75 mm^2 x 2 conductors. It can thus be used with the same specifications as the VCTF cable listed above.

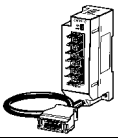
Connectors and Terminal Blocks

Name	Model	Comments
Branch Crimp Connector	SCN1-TH4	Connector used to branch from the main line. Can be used only on the Special Flat Cable.
Extension Crimp Connector	SCN1-TH4E	Used to extend the Special Flat Cable.
Terminating Resistor Crimp Connector	SCN1-TH4T	A connector equipped with terminating resistance. Can be used only on the Special Flat Cable.
Terminal-block Terminator	SRT1-T	A terminal block equipped with terminating resistance. Can be used either on the Special Flat Cable or VCTF cable.

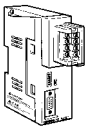
SRM1 RS-232C Port Connecting Cable

Model		Name	Specifications
	CQM1-CIF02	Peripheral Port Conversion Cable	For connecting IBM PC/AT compatible computers. (Cable length: 3.3 m)

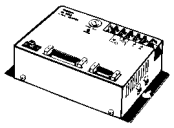
RS-422 Adapter

Model		Name	Specifications
	CPM1-CIF11	RS-422 Adapter	For level conversion between the Peripheral Port and RS-422

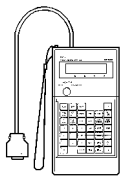
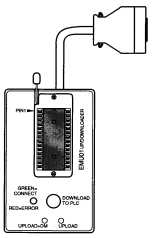
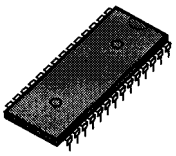
Link Adapters

Model		Name	Specifications
	NT-AL001	Link Adapter	One RS-232C connector and one RS-422 terminal block. Power supply: 5 VDC; 150 mA

Link Adapter for IBM PC/AT-compatible Computers

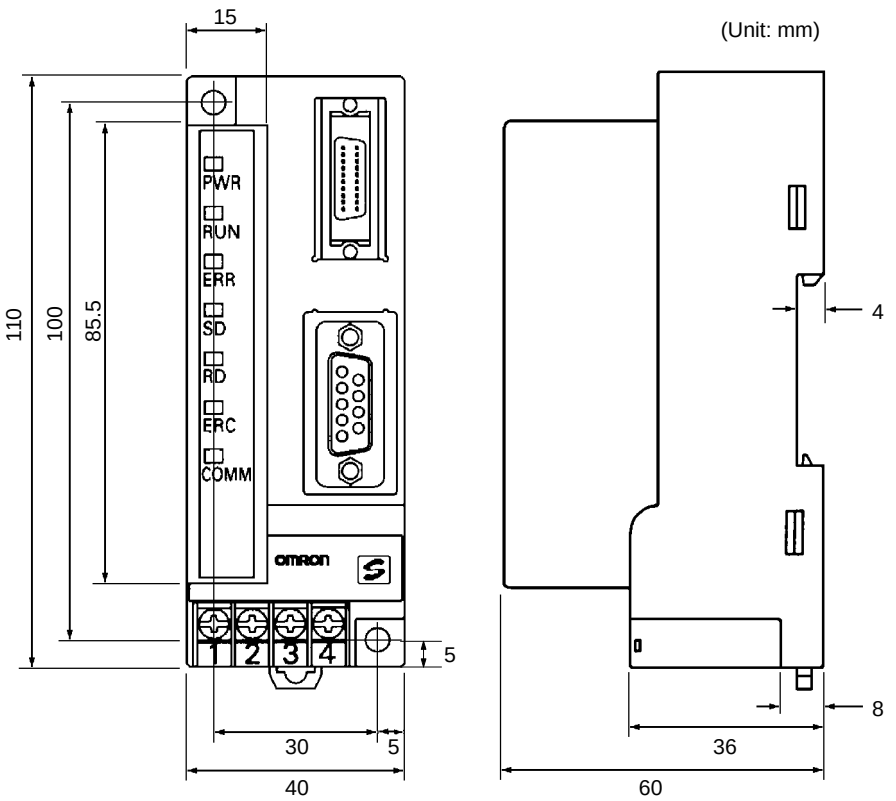
Model		Name	Specifications
	3G2A9-AL004-E	Link Adapter	One RS-232C connector, one RS-422 connector, and one fibre-optic connector. Power supply: 100/200 VAC

Peripheral Devices

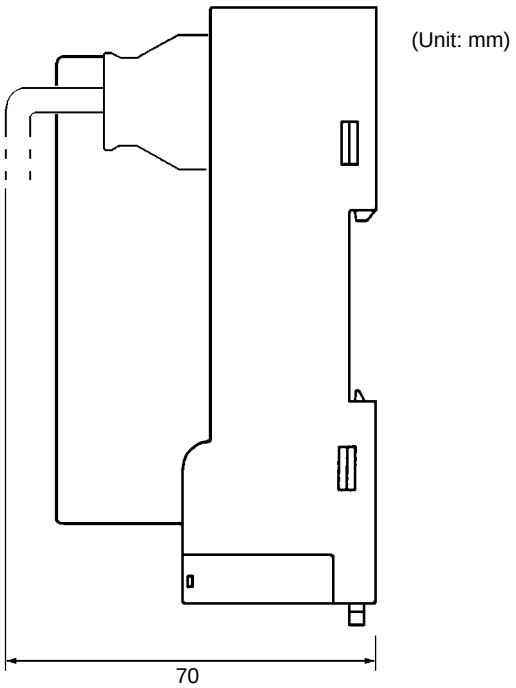
Model		Name	Specifications
	CQM1-PRO01-E	CQM1 Programming Console	With cable (2 m)
	C200H-PRO27-E	C200H Programming Console	Hand-held, with backlight; requires the C200H-CN222 or C200H-CN422, see below.
	C200H-CN222	C200H-PRO27-E Connecting Cable	Cable length: 2 m
	C200H-CN422		Cable length: 4 m
	C200H-ATT01	Mounting Bracket	For panel mounting.
	WS02-CXPC1-E	CX-Programmer	MS-Windows 95/98 (CD-ROM) for Ver. 2.0 or higher
	C500-ZL3AT1-E	SYSMAC Support Software	3.5”, 2HD for IBM PC/AT compatible
	WS01-CPTB1-E	SYSMAC-CPT	For IBM PC/AT or compatible computers (3.5” disks (2HD) and CDROM)
	CPM1-EMU01-V1	Expansion Memory Unit	Uploads the ladder program and DM 6144 to DM 6655 from the PC to the EEPROM and downloads the ladder program and DM 6144 to DM 6655 from the EEPROM to the PC.
	EEROM-JD	EEPROM	256 K bit

Appendix B
External Dimensions

The external dimensions of the SRM1 are as shown in the following diagram.



When mounting a Peripheral Device, be sure to allow sufficient space as shown in the following diagram.



Glossary

***DM** Indirectly addressed DM area. See *indirect address* and *DM area*.

1:1 PC Link A link created between two PCs to create *common data* in their LR areas.

ACP See *add count input*.

add count input An input signal used to increment a counter when the signal changes from OFF to ON.

address A number used to identify the location of data or programming instructions in memory.

AND A logic operation whereby the result is true if and only if both premises are true. In ladder-diagram programming the premises are usually ON/OFF states of bits or the logical combination of such states called execution conditions.

area See *data area* and *memory area*.

area prefix A one or two letter prefix used to identify a memory area in the PC. All memory areas except the IR and SR areas require prefixes to identify addresses in them.

arithmetic shift A shift operation wherein the carry flag is included in the shift.

ASCII Short for American Standard Code for Information Interchange. ASCII is used to code characters for output to printers and other external devices.

AR Area A PC data area allocated to flags and control bits.

AUTOEXEC.BAT An MS DOS file containing commands automatically executed at startup.

back-up A copy made of existing data to ensure that the data will not be lost even if the original data is corrupted or erased.

basic instruction A fundamental instruction used in a ladder diagram. See *advanced instruction*.

baud rate The data transmission speed between two devices in a system measured in bits per second.

BCD See *binary-coded decimal*.

BCD calculation An arithmetic calculation that uses numbers expressed in binary-coded decimal.

binary A number system where all numbers are expressed in base 2, i.e., numbers are written using only 0's and 1's. Each group of four binary bits is equivalent to one hexadecimal digit. Binary data in memory is thus often expressed in hexadecimal for convenience.

binary calculation An arithmetic calculation that uses numbers expressed in binary.

binary-coded decimal A system used to represent numbers so that every four binary bits is numerically equivalent to one decimal digit.

bit The smallest piece of information that can be represented on a computer. A bit has the value of either zero or one, corresponding to the electrical signals ON and OFF. A bit represents one binary digit. Some bits at particular addresses are allocated to special purposes, such as holding the status of input from external devices, while other bits are available for general use in programming.

bit address The location in memory where a bit of data is stored. A bit address specifies the data area and word that is being addressed as well as the number of the bit within the word.

bit designator An operand that is used to designate the bit or bits of a word to be used by an instruction.

bit number A number that indicates the location of a bit within a word. Bit 00 is the rightmost (least-significant) bit; bit 15 is the leftmost (most-significant) bit.

bit-control instruction An instruction that is used to control the status of an individual bit as opposed to the status of an entire word.

block See *logic block* and *instruction block*.

building-block PC A PC that is constructed from individual components, or "building blocks." With building-block PCs, there is no one Unit that is independently identifiable as a PC. The PC is rather a functional assembly of Units.

bus A communications path used to pass data between any of the Units connected to it.

bus bar The line leading down the left and sometimes right side of a ladder diagram. Instruction execution proceeds down the bus bar, which is the starting point for all instruction lines.

byte A unit of data equivalent to 8 bits, i.e., half a word.

call A process by which instruction execution shifts from the main program to a subroutine. The subroutine may be called by an instruction or by an interrupt.

Carry Flag A flag that is used with arithmetic operations to hold a carry from an addition or multiplication operation, or to indicate that the result is negative in a subtraction operation. The carry flag is also used with certain types of shift operations.

central processing unit A device that is capable of storing programs and data, and executing the instructions contained in the programs. In a PC System, the central processing unit executes the program, processes I/O signals, communicates with external devices, etc.

CH See *word*.

channel See *word*.

character code A numeric (usually binary) code used to represent an alphanumeric character.

checksum A sum transmitted with a data pack in communications. The checksum can be recalculated from the received data to confirm that the data in the transmission has not been corrupted.

clock pulse A pulse available at specific bits in memory for use in timing operations. Various clock pulses are available with different pulse widths, and therefore different frequencies.

clock pulse bit A bit in memory that supplies a pulse that can be used to time operations. Various clock pulse bits are available with different pulse widths, and therefore different frequencies.

common data Data that is stored in a memory of a PC and which is shared by other PCs in the same the same system. Each PC has a specified section(s) of the area allocated to it. Each PC writes to the section(s) allocated to it and reads the sections allocated to the other PCs with which it shares the common data.

communications cable Cable used to transfer data between components of a control system and conforming to the RS-232C or RS-422 standards.

comparison instruction An instruction used to compare data at different locations in memory to determine the relationship between the data.

Completion Flag A flag used with a timer or counter that turns ON when the timer has timed out or the counter has reached its set value.

condition A symbol placed on an instruction line to indicate an instruction that controls the execution condition for the terminal instruction. Each condition is assigned a bit in memory that determines its status. The status of the bit assigned to each condition determines the next execution condition. Conditions correspond to LOAD, LOAD NOT, AND, AND NOT, OR, or OR NOT instructions.

CONFIG.SYS An MS DOS file containing environment settings for a personal computer.

constant An input for an operand in which the actual numeric value is specified. Constants can be input for certain operands in place of memory area addresses. Some operands must be input as constants.

control bit A bit in a memory area that is set either through the program or via a Programming Device to achieve a specific purpose, e.g., a Restart Bit is turned ON and OFF to restart a Unit.

control data An operand that specifies how an instruction is to be executed. The control data may specify the part of a word is to be used as the operand, it may specify the destination for a data transfer instructions, it may specify the size of a data table used in an instruction, etc.

control signal A signal sent from the PC to effect the operation of the controlled system.

Control System All of the hardware and software components used to control other devices. A Control System includes the PC System, the PC programs, and all I/O devices that are used to control or obtain feedback from the controlled system.

controlled system The devices that are being controlled by a PC System.

count pulse The signal counted by a counter.

counter A dedicated group of digits or words in memory used to count the number of times a specific process has occurred, or a location in memory accessed through a TIM/CNT bit and used to count the number of times the status of a bit or an execution condition has changed from OFF to ON.

CPU Unit See *central processing unit*.

CTS An acronym for clear-to-send, a signal used in communications between electronic devices to indicate that the receiver is ready to accept incoming data.

CY See *Carry Flag*.

cycle One unit of processing performed by the CPU Unit, including ladder program execution, peripheral servicing, I/O refreshing, etc.

cycle time The time required to complete one cycle of CPU Unit processing.

cyclic interrupt See *scheduled interrupt.*

data area An area in the PC's memory that is designed to hold a specific type of data.

data area boundary The highest address available within a data area. When designating an operand that requires multiple words, it is necessary to ensure that the highest address in the data area is not exceeded.

data disk A floppy disk used to same user programs, DM area contents, comments, and other user data.

data length In communications, the number of bits that is to be treated as one unit in data transmissions.

data link An automatic data transmission operation that allows PCs or Units within PC to pass data back and forth via common data areas.

data link area A common data area established through a data link.

data movement instruction An instruction used to move data from one location in memory to another. The data in the original memory location is left unchanged.

data sharing The process in which common data areas or common data words are created between two or more PCs.

data trace A process in which changes in the contents of specific memory locations are recorded during program execution.

data transfer Moving data from one memory location to another, either within the same device or between different devices connected via a communications line or network.

debug A process by which a draft program is corrected until it operates as intended. Debugging includes both the removal of syntax errors, as well as the fine-tuning of timing and coordination of control operations.

decimal A number system where numbers are expressed to the base 10. In a PC all data is ultimately stored in binary form, four binary bits are often used to represent one decimal digit, via a system called binary-coded decimal.

decrement Decreasing a numeric value, usually by 1.

default A value automatically set by the PC when the user does not specifically set another value. Many devices will assume such default conditions upon the application of power.

definer A number used as an operand for an instruction but that serves to define the instruction itself, rather that the data on which the instruction is to operate. Definers include jump numbers, subroutine numbers, etc.

destination The location where an instruction places the data on which it is operating, as opposed to the location from which data is taken for use in the instruction. The location from which data is taken is called the source.

differentiated instruction An instruction that is executed only once each time its execution condition goes from OFF to ON. Non-differentiated instructions are executed for each scan as long as the execution condition stays ON.

differentiation instruction An instruction used to ensure that the operand bit is never turned ON for more than one scan after the execution condition goes either from OFF to ON for a Differentiate Up instruction or from ON to OFF for a Differentiate Down instruction.

digit A unit of storage in memory that consists of four bits.

digit designator An operand that is used to designate the digit or digits of a word to be used by an instruction.

DIN track A rail designed to fit into grooves on various devices to allow the devices to be quickly and easily mounted to it.

DIP switch Dual in-line package switch, an array of pins in a signal package that is mounted to a circuit board and is used to set operating parameters.

direct output A method in which program execution results are output immediately to eliminate the affects of the cycle time.

distributed control A automation concept in which control of each portion of an automated system is located near the devices actually being controlled, i.e., control is decentralized and ‘distributed’ over the system. Distributed control is a concept basic to PC Systems.

DM area A data area used to hold only word data. Words in the DM area cannot be accessed bit by bit.

DM word A word in the DM area.

downloading The process of transferring a program or data from a higher-level or host computer to a lower-level or slave computer. If a Programming Device is involved, the Programming Device is considered the host computer.

EEPROM Electrically erasable programmable read-only memory; a type of ROM in which stored data can be erased and reprogrammed. This is accomplished using a special control lead connected to the EEPROM chip and can be done without having to remove the EEPROM chip from the device in which it is mounted.

electrical noise Random variations of one or more electrical characteristics such as voltage, current, and data, which might interfere with the normal operation of a device.

EPROM Erasable programmable read-only memory; a type of ROM in which stored data can be erased, by ultraviolet light or other means, and reprogrammed.

error code A numeric code generated to indicate that an error exists, and something about the nature of the error. Some error codes are generated by the system; others are defined in the program by the operator.

Error Log Area An area used to store records indicating the time and nature of errors that have occurred in the system.

even parity A communication setting that adjusts the number of ON bits so that it is always even. See *parity*.

event processing Processing that is performed in response to an event, e.g., an interrupt signal.

exclusive NOR A logic operation whereby the result is true if both of the premises are true or both of the premises are false. In ladder-diagram programming, the premises are usually the ON/OFF states of bits, or the logical combination of such states, called execution conditions.

exclusive OR A logic operation whereby the result is true if one, and only one, of the premises is true. In ladder-diagram programming the premises are usually the ON/OFF states of bits, or the logical combination of such states, called execution conditions.

execution condition The ON or OFF status under which an instruction is executed. The execution condition is determined by the logical combination of conditions on the same instruction line and up to the instruction currently being executed.

execution cycle The cycle used to execute all processes required by the CPU Unit, including program execution, I/O refreshing, peripheral servicing, etc.

execution time The time required for the CPU Unit to execute either an individual instruction or an entire program.

extended counter A counter created in a program by using two or more count instructions in succession. Such a counter is capable of counting higher than any of the standard counters provided by the individual instructions.

extended timer A timer created in a program by using two or more timers in succession. Such a timer is capable of timing longer than any of the standard timers provided by the individual instructions.

FA Factory automation.

factory computer A general-purpose computer, usually quite similar to a business computer, that is used in automated factory control.

FAL error An error generated from the user program by execution of an FAL(06) instruction.

FALS error An error generated from the user program by execution of an FALS(07) instruction or an error generated by the system.

fatal error An error that stops PC operation and requires correction before operation can continue.

FCS See *frame checksum*.

flag A dedicated bit in memory that is set by the system to indicate some type of operating status. Some flags, such as the carry flag, can also be set by the operator or via the program.

flicker bit A bit that is programmed to turn ON and OFF at a specific frequency.

floating-point decimal A decimal number expressed as a number (the mantissa) multiplied by a power of 10, e.g., 0.538×10^{-5}.

force reset The process of forcibly turning OFF a bit via a programming device. Bits are usually turned OFF as a result of program execution.

force set The process of forcibly turning ON a bit via a programming device. Bits are usually turned ON as a result of program execution.

forced status The status of bits that have been force reset or force set.

frame checksum The results of exclusive ORing all data within a specified calculation range. The frame checksum can be calculated on both the sending and receiving end of a data transfer to confirm that data was transmitted correctly.

function code A two-digit number used to input an instruction into the PC.

hardware error An error originating in the hardware structure (electronic components) of the PC, as opposed to a software error, which originates in software (i.e., programs).

header code A code in an instruction that specifies what the instruction is to do.

hexadecimal A number system where all numbers are expressed to the base 16. In a PC all data is ultimately stored in binary form, however, displays and inputs on Programming Devices are often expressed in hexadecimal to simplify operation. Each group of four binary bits is numerically equivalent to one hexadecimal digit.

host computer A computer that is used to transfer data to or receive data from a PC in a Host Link system. The host computer is used for data management and overall system control. Host computers are generally small personal or business computers.

host interface An interface that allows communications with a host computer.

host link An interface connecting a PC to a host computer to enable monitoring or program control from the host computer.

HR area A memory area that preserves bit status during power interrupts and used as work bits in programming.

I/O bit A bit in memory used to hold I/O status. Input bits reflect the status of input terminals; output bits hold the status for output terminals.

I/O capacity The number of inputs and outputs that a PC is able to handle. This number ranges from around one hundred for smaller PCs to two thousand for the largest ones.

I/O delay The delay in time from when a signal is sent to an output to when the status of the output is actually in effect or the delay in time from when the status of an input changes until the signal indicating the change in the status is received.

I/O device A device connected to the I/O terminals on I/O Units. I/O devices may be either part of the Control System, if they function to help control other devices, or they may be part of the controlled system.

I/O interrupt An interrupt generated by a signal from I/O.

I/O point The place at which an input signal enters the PC System, or at which an output signal leaves the PC System. In physical terms, I/O points correspond to terminals or connector pins on a Unit; in terms of programming, an I/O points correspond to I/O bits in the IR area.

I/O refreshing The process of updating output status sent to external devices so that it agrees with the status of output bits held in memory and of updating input bits in memory so that they agree with the status of inputs from external devices.

I/O response time The time required for an output signal to be sent from the PC in response to an input signal received from an external device.

I/O Unit The Units in a PC that are physically connected to I/O devices to input and output signals. I/O Units include Input Units and Output Units, each of which is available in a range of specifications.

I/O word A word in the IR area that is allocated to a Unit in the PC System and is used to hold I/O status for that Unit.

IBM PC/AT or compatible A computer that has similar architecture to, that is logically compatible with, and that can run software designed for an IBM PC/AT computer.

increment Increasing a numeric value, usually by 1.

indirect address An address whose contents indicates another address. The contents of the second address will be used as the actual operand.

initialization error An error that occurs either in hardware or software during the PC System startup, i.e., during initialization.

initialize Part of the startup process whereby some memory areas are cleared, system setup is checked, and default values are set.

input The signal coming from an external device into the PC. The term input is often used abstractly or collectively to refer to incoming signals.

input bit A bit in the IR area that is allocated to hold the status of an input.

input device An external device that sends signals into the PC System.

input point The point at which an input enters the PC System. Input points correspond physically to terminals or connector pins.

input signal A change in the status of a connection entering the PC. Generally an input signal is said to exist when, for example, a connection point goes from low to high voltage or from a nonconductive to a conductive state.

install The preparation necessary to use a program or software package, such as the LSS or SSS, on a computer.

instruction A direction given in the program that tells the PC of the action to be carried out, and the data to be used in carrying out the action. Instructions can be used to simply turn a bit ON or OFF, or they can perform much more complex actions, such as converting and/or transferring large blocks of data.

instruction block — A group of instructions that is logically related in a ladder-diagram program. A logic block includes all of the instruction lines that interconnect with each other from one or more line connecting to the left bus bar to one or more right-hand instructions connecting to the right bus bar.

instruction execution time — The time required to execute an instruction. The execution time for any one instruction can vary with the execution conditions for the instruction and the operands used in it.

instruction line — A group of conditions that lie together on the same horizontal line of a ladder diagram. Instruction lines can branch apart or join together to form instruction blocks. Also called a rung.

interface — An interface is the conceptual boundary between systems or devices and usually involves changes in the way the communicated data is represented. Interface devices perform operations like changing the coding, format, or speed of the data.

interlock — A programming method used to treat a number of instructions as a group so that the entire group can be reset together when individual execution is not required. An interlocked program section is executed normally for an ON execution condition and partially reset for an OFF execution condition.

interrupt (signal) — A signal that stops normal program execution and causes a subroutine to be run or other processing to take place.

interrupt program — A program that is executed in response to an interrupt.

inverse condition — See *normally closed condition*.

JIS — An acronym for Japanese Industrial Standards.

jump — A type of programming where execution moves directly from one point in a program to another, without sequentially executing any instructions in between.

jump number — A definer used with a jump that defines the points from and to which a jump is to be made.

ladder diagram (program) — A form of program arising out of relay-based control systems that uses circuit-type diagrams to represent the logic flow of programming instructions. The appearance of the program is similar to a ladder, and thus the name.

ladder diagram symbol — A symbol used in drawing a ladder-diagram program.

ladder instruction — An instruction that represents the conditions on a ladder-diagram program. The other instructions in a ladder diagram fall along the right side of the diagram and are called terminal instructions.

Ladder Support Software — A software package installed on a IBM PC/AT or compatible computer to function as a Programming Device.

least-significant (bit/word) — See *rightmost (bit/word)*.

LED — Acronym for light-emitting diode; a device used as for indicators or displays.

leftmost (bit/word) — The highest numbered bits of a group of bits, generally of an entire word, or the highest numbered words of a group of words. These bits/words are often called most-significant bits/words.

link — A hardware or software connection formed between two Units. “Link” can refer either to a part of the physical connection between two Units or a software connection created to data existing at another location (i.e., data links).

load The processes of copying data either from an external device or from a storage area to an active portion of the system such as a display buffer. Also, an output device connected to the PC is called a load.

logic block A group of instructions that is logically related in a ladder-diagram program and that requires logic block instructions to relate it to other instructions or logic blocks.

logic block instruction An instruction used to locally combine the execution condition resulting from a logic block with a current execution condition. The current execution condition could be the result of a single condition, or of another logic block. AND Load and OR Load are the two logic block instructions.

logic instruction Instructions used to logically combine the content of two words and output the logical results to a specified result word. The logic instructions combine all the same-numbered bits in the two words and output the result to the bit of the same number in the specified result word.

LR area A data area that is used in data links.

LSS See *Ladder Support Software*.

main program All of a program except for subroutine and interrupt programs.

mark trace A process in which changes in the contents of specific memory locations are recorded during program execution.

masked bit A bit whose status has been temporarily made ineffective.

masking 'Covering' an interrupt signal so that the interrupt is not effective until the mask is removed.

megabyte A unit of storage equal to one million bytes.

memory area Any of the areas in the PC used to hold data or programs.

message number A number assigned to a message generated with the MESSAGE instruction.

mnemonic code A form of a ladder-diagram program that consists of a sequential list of the instructions without using a ladder diagram.

MONITOR mode A mode of PC operation in which normal program execution is possible, and which allows modification of data held in memory. Used for monitoring or debugging the PC.

most-significant (bit/word) See *leftmost (bit/word).*

NC input An input that is normally closed, i.e., the input signal is considered to be present when the circuit connected to the input opens.

negative delay A delay set for a data trace in which recording data begins before the trace signal by a specified amount.

nesting Programming one loop within another loop, programming a call to a subroutine within another subroutine, or programming one jump within another.

NO input An input that is normally open, i.e., the input signal is considered to be present when the circuit connected to the input closes.

noise interference Disturbances in signals caused by electrical noise.

non-fatal error A hardware or software error that produces a warning but does not stop the PC from operating.

normal condition See *normally open condition*.

normally closed condition A condition that produces an ON execution condition when the bit assigned to it is OFF, and an OFF execution condition when the bit assigned to it is ON.

normally open condition A condition that produces an ON execution condition when the bit assigned to it is ON, and an OFF execution condition when the bit assigned to it is OFF.

NOT A logic operation which inverts the status of the operand. For example, AND NOT indicates an AND operation with the opposite of the actual status of the operand bit.

OFF The status of an input or output when a signal is said not to be present. The OFF state is generally represented by a low voltage or by non-conductivity, but can be defined as the opposite of either.

OFF delay The delay between the time when a signal is switched OFF (e.g., by an input device or PC) and the time when the signal reaches a state readable as an OFF signal (i.e., as no signal) by a receiving party (e.g., output device or PC).

offset A positive or negative value added to a base value such as an address to specify a desired value.

ON The status of an input or output when a signal is said to be present. The ON state is generally represented by a high voltage or by conductivity, but can be defined as the opposite of either.

ON delay The delay between the time when an ON signal is initiated (e.g., by an input device or PC) and the time when the signal reaches a state readable as an ON signal by a receiving party (e.g., output device or PC).

one-shot bit A bit that is turned ON or OFF for a specified interval of time which is longer than one scan.

one-to-one link See *1:1 PC Link*.

online edit The process of changed the program directly in the PC from a Programming Device. Online editing is possible in PROGRAM or MONITOR mode. In MONITOR mode, the program can actually be changed while it is being

operand The values designated as the data to be used for an instruction. An operand can be input as a constant expressing the actual numeric value to be used or as an address to express the location in memory of the data to be used.

operand bit A bit designated as an operand for an instruction.

operand word A word designated as an operand for an instruction.

operating modes One of three PC modes: *PROGRAM mode*, *MONITOR mode*, and *RUN mode*.

operating error An error that occurs during actual PC operation as opposed to an initialization error, which occurs before actual operations can begin.

OR A logic operation whereby the result is true if either of two premises is true, or if both are true. In ladder-diagram programming the premises are usually ON/OFF states of bits or the logical combination of such states called execution conditions.

output The signal sent from the PC to an external device. The term output is often used abstractly or collectively to refer to outgoing signals.

output bit A bit in the IR area that is allocated to hold the status to be sent to an output device.

output device An external device that receives signals from the PC System.

output point The point at which an output leaves the PC System. Output points correspond physically to terminals or connector pins.

output signal A signal being sent to an external device. Generally an output signal is said to exist when, for example, a connection point goes from low to high voltage or from a nonconductive to a conductive state.

overflow The state where the capacity of a data storage location has been exceeded.

overseeing Part of the processing performed by the CPU Unit that includes general tasks required to operate the PC.

overwrite Changing the content of a memory location so that the previous content is lost.

parity Adjustment of the number of ON bits in a word or other unit of data so that the total is always an even number or always an odd number. Parity is generally used to check the accuracy of data after being transmitted by confirming that the number of ON bits is still even or still odd.

parity check Checking parity to ensure that transmitted data has not been corrupted.

PC See *Programmable Controller*.

PC configuration The arrangement and interconnections of the Units that are put together to form a functional PC.

PC System With building-block PCs, all of the Units connected up to, but not including, the I/O devices. The boundaries of a PC System are the PC and the program in its CPU Unit at the upper end; and the I/O Units at the lower end.

PCB See *printed circuit board*.

PC Setup A group of operating parameters set in the PC from a Programming Device to control PC operation.

Peripheral Device Devices connected to a PC System to aid in system operation. Peripheral devices include printers, programming devices, external storage media, etc.

peripheral servicing Processing signals to and from peripheral devices, including refreshing, communications processing, interrupts, etc.

port A connector on a PC or computer that serves as a connection to an external device.

positive delay A delay set for a data trace in which recording data begins after the trace signal by a specified amount.

Power Supply Unit A Unit that connected to a PC that provides power at the voltage required by the other Units.

present value The current value registered in a device at any instant during its operation. Present value is abbreviated as PV. The use of this term is generally restricted to timers and counters.

printed circuit board A board onto which electrical circuits are printed for mounting into a computer or electrical device.

PROGRAM mode A mode of operation that allows inputting and debugging of programs to be carried out, but that does not permit normal execution of the program.

Programmable Controller A computerized device that can accept inputs from external devices and generate outputs to external devices according to a program held in memory. Programmable Controllers are used to automate control of external devices. Although single-unit Programmable Controllers are available, building-block Programmable Controllers are constructed from separate components. Such Programmable Controllers are formed only when enough of these separate components are assembled to form a functional assembly.

programmed alarm An alarm given as a result of execution of an instruction designed to generate the alarm in the program, as opposed to one generated by the system.

programmed error An error arising as a result of the execution of an instruction designed to generate the error in the program, as opposed to one generated by the system.

programmed message A message generated as a result of execution of an instruction designed to generate the message in the program, as opposed to one generated by the system.

Programming Console The portable form of Programming Device for a PC.

Programming Device A Peripheral Device used to input a program into a PC or to alter or monitor a program already held in the PC. There are dedicated programming devices, such as Programming Consoles, and there are non-dedicated devices, such as a host computer.

PROM Programmable read-only memory; a type of ROM into which the program or data may be written after manufacture, by a customer, but which is fixed from that time on.

prompt A message or symbol that appears on a display to request input from the operator.

protocol The parameters and procedures that are standardized to enable two devices to communicate or to enable a programmer or operator to communicate with a device.

PV See *present value*.

RAM Random access memory; a data storage media. RAM will not retain data when power is disconnected.

RAS An acronym for reliability, assurance, safety.

read-only area A memory area from which the user can read status but to which data cannot be written.

refresh The process of updating output status sent to external devices so that it agrees with the status of output bits held in memory and of updating input bits in memory so that they agree with the status of inputs from external devices.

relay-based control The forerunner of PCs. In relay-based control, groups of relays are interconnected to form control circuits. In a PC, these are replaced by programmable circuits.

reserved bit A bit that is not available for user application.

reserved word A word in memory that is reserved for a special purpose and cannot be accessed by the user.

reset The process of turning a bit or signal OFF or of changing the present value of a timer or counter to its set value or to zero.

response code A code sent with the response to a data transmission that specifies how the transmitted data was processed.

response format A format specifying the data required in a response to a data transmission.

response monitoring time The time a device will wait for a response to a data transmission before assuming that an error has occurred.

Restart Bit A bit used to restart part of a PC.

result word A word used to hold the results from the execution of an instruction.

retrieve The processes of copying data either from an external device or from a storage area to an active portion of the system such as a display buffer. Also, an output device connected to the PC is called a load.

retry The process whereby a device will re-transmit data which has resulted in an error message from the receiving device.

return The process by which instruction execution shifts from a subroutine back to the main program (usually the point from which the subroutine was called).

reversible counter A counter that can be both incremented and decremented depending on the specified conditions.

reversible shift register A shift register that can shift data in either direction depending on the specified conditions.

right-hand instruction See *terminal instruction*.

rightmost (bit/word) The lowest numbered bits of a group of bits, generally of an entire word, or the lowest numbered words of a group of words. These bits/words are often called least-significant bits/words.

rising edge The point where a signal actually changes from an OFF to an ON status.

ROM Read only memory; a type of digital storage that cannot be written to. A ROM chip is manufactured with its program or data already stored in it and can never be changed. However, the program or data can be read as many times as desired.

rotate register A shift register in which the data moved out from one end is placed back into the shift register at the other end.

RS-232C interface An industry standard for serial communications.

RUN mode The operating mode used by the PC for normal control operations.

rung See *instruction line*.

scan The process used to execute a ladder-diagram program. The program is examined sequentially from start to finish and each instruction is executed in turn based on execution conditions.

scan time See *cycle time*.

scheduled interrupt An interrupt that is automatically generated by the system at a specific time or program location specified by the operator. Scheduled interrupts result in the execution of specific subroutines that can be used for instructions that must be executed repeatedly at a specified interval of time.

SCP See *subtract count input.*

seal See *self-maintaining bit*.

self diagnosis A process whereby the system checks its own operation and generates a warning or error if an abnormality is discovered.

self-maintaining bit A bit that is programmed to maintain either an OFF or ON status until set or reset by specified conditions.

series A wiring method in which Units are wired consecutively in a string.

servicing The process whereby the PC checks a connector or Unit to see if special processing is required.

set The process of turning a bit or signal ON.

set value The value from which a decrementing counter starts counting down or to which an incrementing counter counts up (i.e., the maximum count), or the time from which or for which a timer starts timing. Set value is abbreviated SV.

shift input signal An input signal whose OFF to ON transition causes data to be shifted one bit.

shift register One or more words in which data is shifted a specified number of units to the right or left in bit, digit, or word units. In a rotate register, data shifted out one end is shifted back into the other end. In other shift registers, new data (either specified data, zero(s) or one(s)) is shifted into one end and the data shifted out at the other end is lost.

signed binary A binary value that is stored in memory along with a bit that indicates whether the value is positive or negative.

signed decimal One-word signed hexadecimal values stored in the two's complement format can be displayed at the Programming Console as decimal values from –32,768 to 32,767.

software error An error that originates in a software program.

software protect A means of protecting data from being changed that uses software as opposed to a physical switch or other hardware setting.

source (word) The location from which data is taken for use in an instruction, as opposed to the location to which the result of an instruction is to be written. The latter is called the destination.

special instruction An instruction input with a function code that handles data processing operations within ladder diagrams, as opposed to a basic instruction, which makes up the fundamental portion of a ladder diagram.

SR area A memory area containing flags and other bits/words with specific functions.

SSS See *SYSMAC Support Software*.

store The process of recording a program written into a display buffer permanently in memory.

subroutine A group of instructions placed separate from the main program and executed only when called from the main program or activated by an interrupt.

subroutine number A definer used to identify the subroutine that a subroutine call or interrupt activates.

subtract count input An input signal used to decrement a counter when the signal changes from OFF to ON.

SV See *set value*.

switching capacity The maximum voltage/current that a relay can safely switch on and off.

synchronous execution Execution of programs and servicing operations in which program execution and servicing are synchronized so that all servicing operations are executed each time the programs are executed.

syntax The form of a program statement (as opposed to its meaning).

syntax error An error in the way in which a program is written. Syntax errors can include 'spelling' mistakes (i.e., a function code that does not exist), mistakes in specifying operands within acceptable parameters (e.g., specifying read-only bits as a destination), and mistakes in actual application of instructions (e.g., a call to a subroutine that does not exist).

SYSMAC Support Software A software package installed on a IBM PC/AT or compatible computer to function as a Programming Device.

system configuration The arrangement in which Units in a System are connected. This term refers to the conceptual arrangement and wiring together of all the devices needed to comprise the System.

system error An error generated by the system, as opposed to one resulting from execution of an instruction designed to generate an error.

system error message An error message generated by the system, as opposed to one resulting from execution of an instruction designed to generate a message.

system setup Operating environment settings for a Programming Device, e.g., the LSS or SSS.

terminal instruction An instruction placed on the right side of a ladder diagram that uses the final execution conditions of an instruction line.

timer A location in memory accessed through a TIM/CNT bit and used to time down from the timer's set value. Timers are turned ON and reset according to their execution conditions.

TR area A data area used to store execution conditions so that they can be reloaded later for use with other instructions.

TR bit A bit in the TR area.

trace An operation whereby the program is executed and the resulting data is stored to enable step-by-step analysis and debugging.

trace memory A memory area used to store the results of trace operations.

transfer The process of moving data from one location to another within the PC, or between the PC and external devices. When data is transferred, generally a copy of the data is sent to the destination, i.e., the content of the source of the transfer is not changed.

transmission distance The distance that a signal can be transmitted.

trigger A signal used to activate some process, e.g., the execution of a trace operation.

trigger address An address in the program that defines the beginning point for tracing. The actual beginning point can be altered from the trigger by defining either a positive or negative delay.

UM area The memory area used to hold the active program, i.e., the program that is being currently executed.

Unit In OMRON PC terminology, the word Unit is capitalized to indicate any product sold for a PC System. Most of the names of these products end with the word Unit.

unit number A number assigned to some Units to facilitate identification when assigning words or other operating parameters.

unmasked bit A bit whose status is effective. See *masked bit*.

unsigned binary A binary value that is stored in memory without any indication of whether it is positive or negative.

unsigned decimal One-word hexadecimal values can be displayed at the Programming Console as decimal values from 0 to 65,535.

uploading The process of transferring a program or data from a lower-level or slave computer to a higher-level or host computer. If a Programming Devices is involved, the Programming Device is considered the host computer.

watchdog timer A timer within the system that ensures that the scan time stays within specified limits. When limits are reached, either warnings are given or PC operation is stopped depending on the particular limit that is reached.

WDT See *watchdog timer*.

word A unit of data storage in memory that consists of 16 bits. All data areas consists of words. Some data areas can be accessed only by words; others, by either words or bits.

word address The location in memory where a word of data is stored. A word address must specify (sometimes by default) the data area and the number of the word that is being addressed.

work area A part of memory containing work words/bits.

work bit A bit in a work word.

work word A word that can be used for data calculation or other manipulation in programming, i.e., a 'work space' in memory. A large portion of the IR area is always reserved for work words. Parts of other areas not required for special purposes may also be used as work words.

write protect switch A switch used to write-protect the contents of a storage device, e.g., a floppy disk. If the hole on the upper left of a floppy disk is open, the information on this floppy disk cannot be altered.

write-protect A state in which the contents of a storage device can be read but cannot be altered.

Index

A–B

accessibility, 20

allocations
  data areas, 9
  I/O, 8

ASCII, converting displays, Programming Console, 57

binary data, modifying, Programming Console, 54

bit status, force-set/reset, Programming Console, 56

bits, searching, Programming Console, 42

buzzer operation, Programming Console, 39

C

cabinet
  installation, precautions, 19
  Programming Console installation, 35

cable connections
  NT Link
    1:1, 30
    1:N, 31
  RS-232C, 24

characteristics, 13

check levels, program checks, 75

checking, program syntax, Programming Console, 47, 71

clearing, memory areas, Programming Console, 37, 62

communications
  CompoBus/S specifications, 14
  errors, 72
  Host Link connections, 26
  modes, 6
  NT Link, 30
  PC Link, 1:1, 31

CompoBus/S
  communications specifications, 14
  wiring, 23

components, Units, 15

conduit installation, 22

configuration, 3

counters, changing SV, Programming Console, 52

crimp connectors, 22

cycle time, displaying, Programming Console, 58

D

data, modifying, Programming Console, 52, 53, 54, 55, 56

data areas, 9

data link, 31

decimal data with sign. *See* signed decimal data

decimal data without sign. *See* unsigned decimal data

differentiated instructions, entering, 45

dimensions, 97

DIN track, installation, 20

displays
  converting between hex and ASCII, Programming Console, 57
  cycle time, Programming Console, 58

downloading, programs, 89

duct installation, 21

E

EEPROM
  mounting/removing, 86
  specifications, 85

electrical noise, preventing, 19, 21

ERR/ALM indicator
  flashing, 72
  lit, 73

errors
  communications, 72
  error processing, 59
  fatal, 72
  identifying, 73
  non-fatal, 72
  programming, 75
  Programming Console operations, 75
  reading/clearing messages, Programming Console, 38, 74
  user-defined errors, 73

expansion instructions, setting, Programming Console, 40–41

Expansion Memory Unit, 83
  connections, 87

F

FAL numbers, 72

FAL(06), 73

FALS numbers, 72

FALS(07), 74

fatal errors, troubleshooting, 79

features, 2

flash memory, precautions, 61

force-set/reset
  clearing, Programming Console, 57
  Programming Console, 56

functions, 2

H

hexadecimal data, converting displays, Programming Console, 57

high-speed communications mode, 6

host computer, connecting, 26

Host Link, connections, 26

Host Link communications
  1:1, 27
    cables, 27
  1:N, 28
    cables, 29

I

I/O allocation, 8

I/O line noise, preventing, 21

IBM PC/AT or compatible computer, 7

indicators, 16
  flashing ERR/ALM indicator, 72
  lit ERR/ALM indicator, 73

installation
  DIN track, 20
  horizontal surface, 20
  precautions, 19

instructions
  inserting and deleting, Programming Console, 43
  searching, Programming Console, 41

interlock circuit, example, 18

interval timer, 2

L–M

limit circuit, 18

lithium battery, backup time vs. temperature, xv, 14

long-distance communications mode, 6

memory
  backup, 2
  Expansion Memory Unit, 83

memory areas
  clearing, Programming Console, 37, 62
  partial clear, 38, 62
  uploading/downloading, 84

messages, reading/clearing, 38, 74

mode, changing the PC mode, 35

models, SRM1 models, 4

modifying
  binary data, Programming Console, 54
  data, Programming Console, 52
  hexadecimal/BCD data, Programming Console, 53
  signed decimal data, Programming Console, 55
  SV, Programming Console, 52
  unsigned decimal data, Programming Console, 56

MONITOR mode, description, 36

monitoring
  3-word monitor, Programming Console, 50–51
  binary monitor, Programming Console, 50
  differentiation monitor, Programming Console, 49–50
  signed decimal monitor, Programming Console, 51
  status, Programming Console, 47
  unsigned decimal monitor, Programming Console, 51–52

MSG(46), 74

N

noise, preventing electrical noise, 19, 21

non-fatal errors, troubleshooting, 80

NT Link
  1:1, 30
  1:N, 30

NT Link communications
  1:1, 30
  1:N, 30
  cable connections, 30, 31

O–P

operating conditions, troubleshooting, 82

overheating, 19

panel, Programming Console installation, 35

panel installation, precautions, 19

PC Link communications, 1:1, 31
  cable connections, 32
  restrictions, 32

PC mode, changing, 35

Peripheral Devices, connecting, 6

peripheral port, 15

peripherals, 5
  IBM PC/AT or compatible computer, 7
  Programming Consoles, 6
  PT, 7
  Slaves, 5
  SYSMAC Support Software, 6
  SYSMAC-CPT, 6

power failure, 18
  momentary, 18

power supply, 18
  troubleshooting, 78
  wiring, 18, 23

precautions
  general, xi
  uploading/downloading, 84

program memory, setting address and reading content, Programming Console, 41

PROGRAM mode, description, 36

Programmable Terminal. *See* PT

programming
  checks for syntax, Programming Console, 47, 71–72
  errors, 75
  inserting and deleting instructions, Programming Console, 43–44
  searching, Programming Console, 41–42
  setting and reading a memory address, Programming Console, 41

Programming Console
  connecting, 6, 35
  keys, 34
  models, 34
  operations, 36

programs
  checking, check levels, 75
  downloading, 89
  entering and editing, Programming Console, 44
  uploading, 88

PT, 2, 7

R–S

RS-232C
  cable connections, 24
  wiring, 24

RS-232C Adapter, 7, 24

RS-232C port, 15

RUN mode, description, 36

scheduled interrupts, 2

search
  instruction, Programming Console, 41–42
  operands, Programming Console, 42

signed decimal data
  modifying, Programming Console, 55
  monitoring, 51

Slaves, list, 5

specifications, 12
  CompoBus/S communications, 14
  Expansion Memory Unit, 85

SSS. *See* SYSMAC Support Software

startup procedure, 60

status, monitoring, Programming Console, 47

SV, modifying, Programming Console, 52

syntax, checking the program, Programming Console, 47, 71

SYSMAC Support Software, connecting, 6

SYSMAC-CPT, 6, 7
  connecting, 6

system, configuration, 3

T–W

temperature, effect on capacitor backup, 14

terminal block, 15

test run, procedure, 60, 71

timers, changing SV, Programming Console, 52

troubleshooting, 77
  fatal errors, 79
  non-fatal errors, 80
  operating conditions, 82
  power supply, 78

UM area information, reading, Programming Console, 40

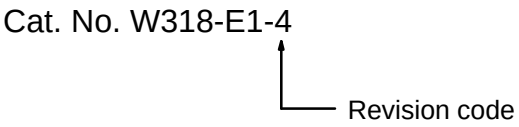
unsigned decimal data
  modifying, Programming Console, 56
  monitoring, 51–52

uploading, programs, 88

wiring
  CompoBus/S, 23
  power supply, 23
  precautions, 21
  RS-232C, 24

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.



The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	April 1997	Original production
2	November 1997	Corrections and modifications resulting from the upgrade to “-V1.” Information on the SYSMAC-CPT Support Software added throughout the manual. **Page 41:** Second operand display for step 10 corrected.
3	April 1999	Corrections and modifications resulting from the upgrade to “-V2.”
4	May 2000	Added *Section 6 Expansion Memory Unit.* Updated *Precautions.* **Page 3:** Added information on the CPM1-EMU01-V1 Expansion Memory Unit. **Page 12:** Removed “1G” and “15G” from the table. **Pages 38 to 58, 62, 63, and 71:** Corrected contents of displays. **Page 83:** Removed notes from the table. **Page 86:** Added models related to the CX-Programmer and Expansion Memory Unit.